LOGIC Emulation Source

# DN8000K10PCI User Guide

LOGIC EMULATION SOURCE

# DN8000K10PCI User Manual Revision 3

February 2, 2009 6:05 PM

---

© The Dini Group, 2005

7469 Draper Ave.
La Jolla, CA92037
Phone 858.454.3419 • Fax 858.454.1728
support@dinigroup.com
www.dinigroup.com

---

# Table of Contents


# ABOUT THIS MANUAL ..... 8

**1 MANUAL CONTENTS ..... 9**

About This Manual ..... 9
Quick Start Guide ..... 9
Board Hardware ..... 9
Controller Software ..... 9
Reference Design ..... 9
FPGA Design Guide ..... **Error! Bookmark not defined.**
Ordering Information ..... 9

**2 ADDITIONAL RESOURCES ..... 9**

**3 CONVENTIONS ..... 10**

3.1 Typographical ..... 10
3.2 Content ..... 11
*3.2.1 File names ..... 11*
*3.2.2 Physical orientation and Origin ..... 11*
*3.2.3 Part Pin Names ..... 12*
*3.2.4 Schematic Clippings ..... 12*
*3.2.5 Terminology ..... 12*

# QUICK START GUIDE ..... 14

**1 PROVIDED MATERIALS ..... 14**

**2 ESD WARNING ..... 15**

**3 POWER-ON INSTRUCTIONS ..... 15**

3.1 Verify Jumper Setting ..... 17
3.2 Verify Switch Settings ..... 17
3.3 Memory and heatsinks ..... 19
3.4 Prepare configration files ..... 19
3.5 Connect cables ..... 20
3.6 View configuration feedback over RS232 ..... 21
*3.6.1 Watch the configuration status output ..... 21*
*3.6.2 Interactive configuration ..... 24*
*3.6.3 Read temperature sensors ..... 24*
*3.6.4 Multiplex Serial port ..... 25*
3.7 Check LED status lights ..... 25

**4 USING THE REFERENCE DESIGN WITH THE PROVIDED SOFTWARE ..... 30**

4.1 Operating the USB controller program ..... 30
4.2 Communicating to the User Design over the Serial Port ..... 32
4.3 Using AETEST to run hardware tests ..... 33
*4.3.1 AETest, Windows98 version ..... 33*
*4.3.2 AETest, DOS version ..... 33*
*4.3.3 AETest on Linux or Solaris ..... 33*
*4.3.4 Use AETest ..... 34*
4.4 Moving On ..... 35

# CONTROLLER SOFTWARE ..... 36

**1 USB CONTROLLER ..... 36**

1.1 MENU OPTIONS ..... 36
*1.1.1 File Menu ..... 36*
*1.1.2 Edit Menu ..... 37*
*1.1.3 FPGA Configuration Menu ..... 37*
*1.1.4 Settings/Info Menu ..... 38*

**2 PCI AETEST APPLICATION ..... 41**

2.1 FUNCTIONALITY ..... 41
2.2 RUNNING AETEST ..... 42
2.3 COMPILING AETEST ..... 42
*2.3.1 Compiling AETest for DOS ..... 42*
*2.3.2 Compiling AETest for Windows XP ..... 43*

**3 UPDATING THE FIRMWARE ..... 43**

3.1 UPDATING THE MCU (FLASH) FIRMWARE ..... 43
3.2 UPDATING THE SPARTAN (EEPROM) FIRMWARE ..... 45

# HARDWARE ..... 49

**4 OVERVIEW ..... 49**

**5 CONFIGURATION CIRCUIT ..... ERROR! BOOKMARK NOT DEFINED.**

5.1 OVERVIEW ..... 50
5.2 THE SPARTAN 2 FPGA ..... 51
*5.2.1 Spartan Configuration ..... 52*
*5.2.2 Smart Media ..... 53*
*5.2.3 MCU communication ..... 54*
*5.2.4 PCI commnication ..... 54*
*5.2.5 RS232 ..... 54*
*5.2.6 IIC ..... 55*
5.3 CONFIGURATION OPTIONS ..... 55
*5.3.1 Jtag ..... 56*
*5.3.2 SmartMedia ..... 57*
*5.3.3 USB ..... 59*
*5.3.4 PCI ..... 60*
5.4 FPGA CONFIGURATION PROCESS ..... 62
5.5 MCU ..... 63
*5.5.1 RS232 ..... 64*
*5.5.2 Clocks ..... 64*
*5.5.3 LEDs ..... 65*
*5.5.4 Memory space ..... 65*
*5.5.5 USB ..... 68*
*5.5.6 Smart media ..... 69*

**6 CLOCKING ..... 70**

6.1 GLOBAL CLOCKS ..... 73
6.2 USER CLOCK ..... 77
6.3 FEEDBACK CLOCKS ..... 78

**7 RESET TOPOLOGY ..... 78**

**8 POWER ..... 80**

8.1 SWITCHING POWER SUPPLIES ..... 82
8.2 SECONDARY POWER SUPPLIES ..... 83
*8.2.1 DDR2 Termination Power ..... 83*

*8.2.2 RocketIO power* ..... *84*
*8.2.3 Optical Module Power* ..... *84*
8.3 HEAT DISSIPATION ..... 85

**9 FPGA INTERCONNECT ..... 86**

9.1 MAIN BUS ..... 89

**10 MEMORY INTERFACE ..... 89**

10.1 CLOCKING ..... 90
10.2 SERIAL PRESENCE DETECT. ..... 90

**11 HEADERS ..... 91**

11.1 3000K10 COMPATIBILITY ..... 93
11.2 FPGA CONNECTION ..... 93
11.3 IO POWER ..... 98
11.4 PHYSICAL ..... 99
11.5 DAUGHTERCARD POWER ..... 99
11.6 THE MICTOR ..... 100

**12 LEDS ..... 102**

**13 ROCKETIO ..... 103**

13.1 ROCKETIO CLOCK RESOURCES ..... 103
13.2 MGT POWER NETWORK ..... 106
*13.2.1 FX CES2 rework ..... 106*
13.3 THE CONNECTIONS ..... 107
13.4 SAMTEC MULTI GIGABIT RIBBON CABLE ..... 110
13.5 OPTICAL MODULES ..... 112
*13.5.1 SFP ..... 112*
*13.5.2 XFP ..... 113*
13.6 THE SMAS ..... 115

**14 PCI INTERFACE ..... 115**

14.1 PCI EDGE CONNECTOR ..... 115
14.2 THE QUICKLOGIC 5064 ..... 116
14.3 VIRTEX 4 FPGA COMMUNICATION ..... 116
14.4 SPARTAN 2 COMMUNICATION ..... 116
14.5 PCI CLOCKING ..... 118
14.6 JTAG ..... 118
14.7 PCI POWER ..... 118
14.8 PCI SIGNALING ..... 119

**15 FPGA SYSTEM MONITOR/ADC ..... 120**

**16 MECHANICAL ..... ERROR! BOOKMARK NOT DEFINED.**

**1 EXPLORING THE REFERENCE DESIGN ..... 122**

1.1 WHAT IS THE REFERENCE DESIGN? ..... 122

**2 REFERENCE DESIGN MEMORY MAP ..... 123**

2.1 USING THE REFERENCE DESIGN ..... 126
*2.1.1 Built-In RocketIO test ..... 127*

**3 MEMORY MAPPED DATA FLOW ..... ERROR! BOOKMARK NOT DEFINED.**

3.1 COMPILING THE REFERENCE DESIGN ..... 133
*3.1.1 The Xilinx Embedded Development Kit (EDK) ..... 133*
*3.1.2 Xinlinx XST ..... **Error! Bookmark not defined.***
*3.1.3 Xilinx ISE ..... 134*
*3.1.4 The Build Utility: Make.bat ..... 134*

**4 GETTING MORE INFORMATION ..... 134**

4.1 PRINTED DOCUMENTATION ..... 138

4.2 ELECTRONIC DOCUMENTATION ..... 138
4.3 ONLINE DOCUMENTATION ..... 138

**5 FPGA OPTIONS ..... 141**

5.1 FPGA A: ..... 141
5.2 FPGA B: ..... 141
5.3 FPGA C: ..... 142

**6 MULTI-GIGABIT SERIAL OPTIONS ..... 142**

6.1 SERIAL CLOCK CRSTALS ..... 142
6.2 MODULE SOCKETS ..... 142

**7 OTHER OPTIONS ..... 143**

7.1 3.3 V HEADERS ..... 143
7.2 12V POWER ..... 143

**8 OPTIONAL EQUIPMENT ..... 143**

Chapter

1

# About This Manual

*Welcome to DN8000K10PCI Logic Emulation Board*

*Congratulations on your purchase of the DN8000K10PCI LOGIC Emulation Board. If you are unfamiliar with Dini Group products, you should read Chapter 2, Quick Start Guide to familiarize yourself with the user interfaces the DN8000K10PCI provides.*

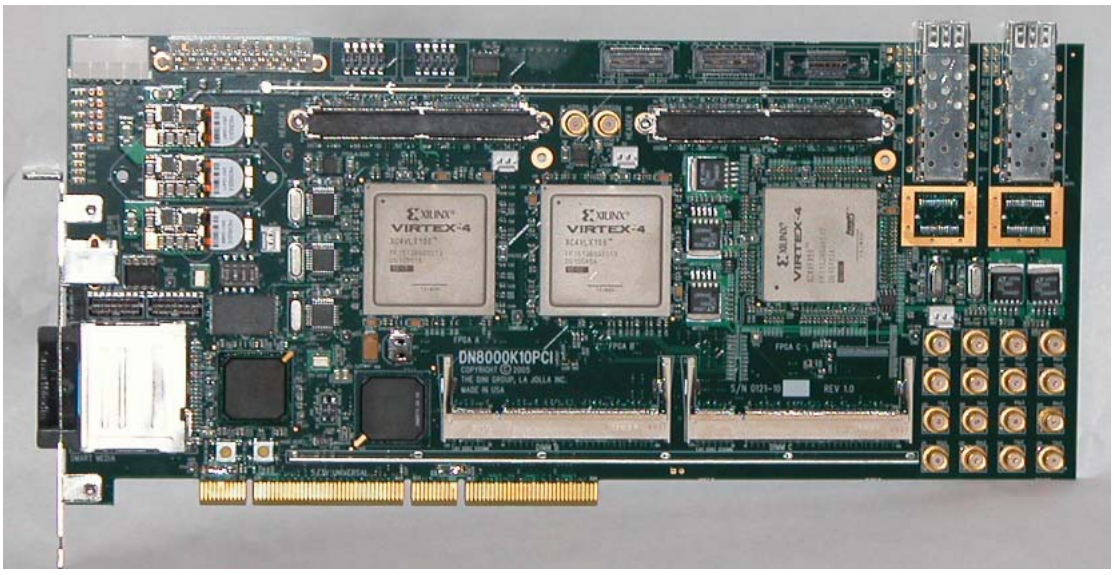



Figure 1 DN8000K10PCI

# 1 Manual Contents

This manual contains the following chapters:

**About This Manual**
List of available documentation and resources available. Reader's Guide to this manual

**Quick Start Guide**
Step-by-step instructions for powering on the DN8000K10PCI, loading and communicating with a simple provided FPGA design and using the board controls.

**Board Hardware**
Detailed description and operating instructions of each individual circuit on the DN8000K10PCI

**Controller Software**
A summary of the functionality of the provided software. Implementation details for the remote USB board control functions and instructions for developing your own USB host software.

**Reference Design**
Detailed description of the provided DN8000K10PCI reference design. Implementation details of the reference design interaction with DN8000K10PCI hardware features.

**Ordering Information**
Contains a list of the available options and available optional equipment. Some suggested parts and equipment available from third party vendors.

# 2 Additional Resources

For additional information, go to http://www.dinigroup.com. The following table lists some of the resources you can access from this website. You can also directly access these resources using the provided URLs.

| Resource | Description/URL |
|---|---|
| UserDN8000K10PCI User Guide | This is the main source of technical information. The manual should contain most of the answers to your questions |
| Dini Group Web Site | The web page will contain the latest manual, application notes, FAQ, articles, and any device errata and manual addenda. Please visit and bookmark: http://www.dinigroup.com |

| Resource | Description/URL |
|---|---|
| Virtex 4 User Guide | Xilinx publication UG070<br><br>http://www.xilinx.com/bvdocs/userguides/ug070.pdf<br><br>Most of your questions regarding usage and capabilities of the Virtex 4 devices will be answered here, including readback, boundary scan, configuration, and debugging |
| E-Mail | You may direct questions and feedback to the Dini Group using this e-mail address: support@dinigroup.com |
| Phone Support | Call us at **858.454.3419** during the hours of 8:00am to 5:00pm Pacific Time. |
| FAQ | The download section of the web page contains a document called **DN8000K10PCI Frequently Asked Questions (FAQ)**. This document is periodically updated with information that may not be in the User's Manual. |

Figure 2 Support Resources

# 3 Conventions

This document uses the following conventions. An example illustrates each convention.

## 3.1 Typographical

The following typographical conventions are used in this document:

| Convention | Meaning or Use | Example |
|---|---|---|
| `Courier font` | Messages, prompts, and program files that the system displays | `speed grade: -100` |
| **`Courier bold`** | Literal commands that you enter in a syntactical statement | **`ngdbuild design_name`** |
| **Garamond bold** | Commands that you select from a menu | **File →Open** |
| | Keyboard shortcuts | **Ctrl+C** |
| | Variables in a syntax statement for which you must supply values | ngdbuild *design_name* |

| Convention | Meaning or Use | Example |
|---|---|---|
| *Italic font* | References to other manuals | See the *Development System Reference Guide* for more information. |
| | Emphasis in text | If a wire is drawn so that it overlaps the pin of a symbol, the two nets are *not* connected. |
| Braces [ ] | An optional entry or parameter. However, in bus specifications, such as bus[7:0], they are required. | ngdbuild [*option_name*] *design_name* |
| Braces { } | A list of items from which you must choose one or more | **lowpwr =**{**on**\|**off**} |
| Vertical bar \| | Separates items in a list of choices | **lowpwr =**{**on**\|**off**} |
| Vertical ellipsis<br>.<br>.<br>. | Repetitive material that has been omitted | IOB #1: Name = QOUT'<br>IOB #2: Name = CLKIN'<br>.<br>. |
| Horizontal ellipsis . . . | Repetitive material that has been omitted | **allow block** *block_name*<br>*loc1 loc2 ... locn;* |
| Prefix "0x" or suffix "h" | Indicates hexadecimal notation | Read from address 0x00110373, returned 4552494h |
| Letter "#" or "_n" | Signal is active low | INT# is active low<br>fpga_inta_n is active low |

## 3.2 Content

### 3.2.1 File names

Paths to documents included on the User CD are prefixed with "D:\". This refers to your CD drive's root directory, or the directory where you have copied the CD contents.

### 3.2.2 Physical orientation and Origin

By convention, the board is oriented as show on page 3, with the "top" of the board being the edge near Headers A and B, and the edge with the optical module connectors. The "right" edge is near the SMA connectors, the "left" side is the side with the PCI bezel. "topside" refers to the

side of the PWB with FPGAs soldered to it, "backside" is the side with the daughter card connectors. The reference origin of the board is the center of the lower PCI bezel mounting hole.

### 3.2.3 Part Pin Names

Pin names are given in the form <*X*><*Y*>.<*Z*>; The <*X*> is one of: U for ICs, R for resistors, C for capacitors, P or J for connectors, FB or L for inductors, TP for test points, MH for mounting structures, FD for fiducials, BT for sockets, DS for diodes, F for fuses, HS for mechanicals, PSU for power supply modules, Q for discreet semiconductors, RN for resistor networks, X for oscillators, Y for crystals. <*Y*> is a number uniquely identifying each part from other parts of the same X class on the same PWB. <*Z*> is the pin or terminal number or name, as defined in the datasheet of the part. Datasheets for all standard and optional parts used on the DN8000K10PCI are included in the Document library on the provided User CD.

### 3.2.4 Schematic Clippings

Partial schematic drawings are included in this document to aid quick understanding of the features of the DN8000K10PCI. These clippings have been modified for clarity and brevity, and may be missing signals, parts, net names and connections. Unmodified Schematics are included in the User CD. Please refer to this document. Use the PDF search feature to search for nets and parts.
D:\Schematics\REV3\Schematic_DN8000K10PCI_503-0121-0000_rev03.pdf

### 3.2.5 Terminology

Abbreviations and pronouns are used for some commonly used phrases.

**MGT** and **RocketIO** are used interchangeably. MGT is multi-gigabit transceiver. RocketIO is the Xilinx trademark on their multi gigabit transceiver hardware.

**MCU** is the Cypress FX2 Microcontroller, U39

Chapter

2

# Quick Start Guide

*The Dini Group DN8000K10PCI is the user-friendliest board available with multiple Virtex 4 FPGAs. However, due to the number of features and flexibility of the board, it will take some time to become familiar with all the control and monitoring interfaces equipped on the DN8000K10PCI. Please follow this quick start guide to become familiar with the board before starting your ASIC emulation project.*

## 1 Provided Materials

Examine the contents of your DN8000K10PCI kit. It should contain:

- DN8000K10PCI board
- Two Smart Media cards
- USB SmartMedia card reader
- RS232 IDC header cable to female DB9
- USB cable
- CD ROM containing:
  - Virtex 4 Reference Design
  - Virtex 4 Documentation (Datasheets)
  - User manual PDF
  - Board Schematic PDF
  - Board netlist TXT
  - USB Windows program (usbcontroller.exe)
  - USB Windows documentation
  - PCI program (Aetestdj.exe)
  - Source code for USB program, PCI program and DN8000K10PCI firmware

You should use the customer netlist provided on the user CD for the purpose of generating designs and should consult the schematic PDF provided on the CD when designing any hardware to interface to the DN8000K10PCI. All other documentation materials are derived from this schematic.

For information about the USB Controller program, see the document accompanying the program on the CD.

# 2 ESD Warning

The DN8000K10PCI is sensitive to static electricity, so treat the PCB accordingly. The target markets for this product are engineers that are familiar with FPGAs and circuit boards. However, if needed, the following web page has an excellent tutorial on the "Fundamentals of ESD" for those of you who are new to ESD sensitive products:

http://www.esda.org/basics/part1.cfm

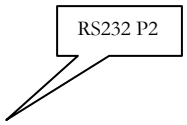
There are two large grounded metal rails on the DN8000K10PCI.

The DN8000K10PCI has been factory tested and pre-programmed to ensure correct operation. reference design is included on the provided CD and SmartMedia card.

The 200-pin connectors are not 5V tolerant. According to the Virtex 4 datasheets, the maximum applied voltage to these signals is VCCO + 0.5V (3.0V while powered on). These connections are not buffered, and the Virtex 4 part is sensitive to ESD. Take care when handling the board to avoid touching the daughter card connectors.

# 3 Power-On Instructions

The image below represents your DN8000K10PCI. You will need to know the location of the following parts referenced in this chapter.



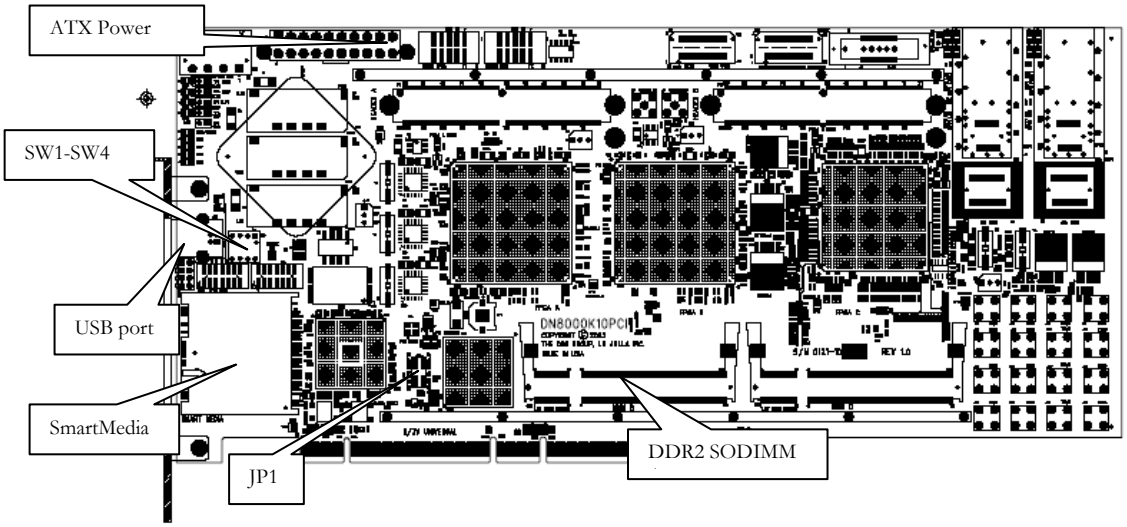



Figure 3 DN8000K10PCI configuration controls

To begin working with the DN8000K10PCI, follow the steps below :

## 3.1 Verify Jumper Setting

The DN8000K10PCI uses a jumper to configure the PCI interface IO signal levels. Please locate the jumper block on your board and verify that it matches the default settings. By default, this jumper is installed from JP1.2-JP1.4

Table 1 – describes the functionality of the jumper block on the DN8000K10PCI.

Table 1 – Jumper Description

| Jumper Location | Signal Name | Description |
|---|---|---|
| JP1.1 – JP1.3 | BOARD 5V | This jumper fixes the PCI clamping voltage at 5.0V. You should use this jumper setting if the DN8000K10PCI is being used in stand-alone mode |
| JP1.2 – JP1.4 (default) | PCI VIO | This jumper should be installed when the board is being operated in a 5V or 3.3V PCI slot. |
| | | **Don't install both jumpers at the same time.** |

The "top" two pins are 1 and 3. Pin 1 is the one with a white circle next to it. There is a label next to each pin indicating it's connection:

1 - BOARD 5V
2 - BOARD VIO
3 - BOARD VIO
4 - PCI VIO

BOARD 5V - BOARD VIO (1-3) is correct when the board is not in PCI. PCI VIO - BOARD VIO (2-4) is correct when in a PCI slot.

The effect of an incorrect setting when in PCI is less reliability in a very heavily loaded PCI slot at 66 Mhz. The effect of the incorrect setting when not in a PCI slot is the temperature sensors might not work.

## 3.2 Verify Switch Settings

The DN8000K10PCI uses a DIP switch to program the FPGA configuration circuitry. The function of these DIP switches is Listed in Table 2. Verify that the switch settings on your board match the default settings.

Table 2 – Switch Description

| Switch | Default Position | Signal Name | On setting | Off setting |
|---|---|---|---|---|
| S1-1 | Off | Reserved | | |
| S1-2 | Off | Reserved | | |
| S1-3 | Off | Boot mode | Firmware update | Normal operation (default) |
| S1-4 | Off | Reserved | | |

## 3.3 Memory and heatsinks

There should be an active heatsink installed on each FPGA on the DN8000K10PCI and a fan over the power supply units. Virtex 4 FPGAs are capable of dissipating 15W or more, so you should always run them with heatsinks installed.

The DN8000K10PCI comes packaged without memory installed. If you want the Dini Group reference design to test your memory modules, you can install them now in the 1.8V DDR2 DIMM sockets.

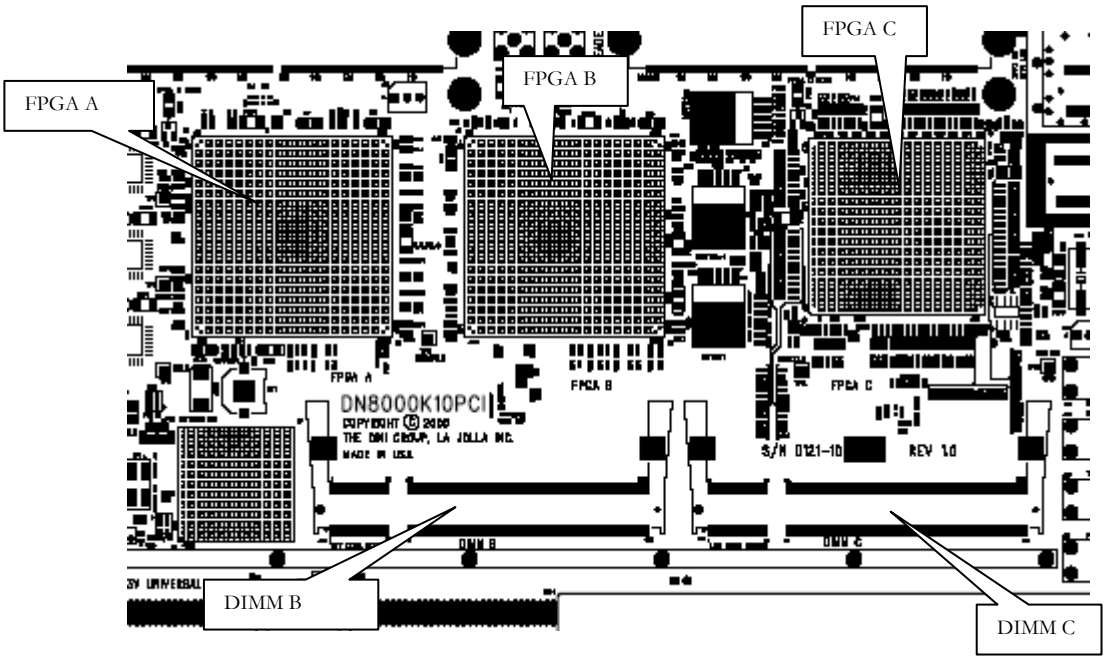



Figure 4 FPGA Names

The socket DIMMB is connected to FPGA B. The socket can accept any capacity DDR2 SODIMM module. Note that DDR1 modules will not work in these slots since they are supplied with 1.8V power and DDR1 requires 2.5V power (and a completely different pin-out).

## 3.4 Prepare configuration files

The DN8000K10PCI reads FPGA configuration data from a SmartMedia card. To program the FPGAs on the DN8000K10PCI, FPGA design files (with a .bit file extension) put on the root directory of the SmartMedia card file using the provided USB card reader.

The DN8000K10PCI ships with a 32 MB SmartMedia card preloaded with the Dini Group reference design.

1. Insert the provided SmartMedia card labeled "Reference Design" into your USB card reader. Make sure the card contains the files:

   FPGA_A.bit

   FPGA_B.bit

   FPGA_C.bit

   main.txt

   The files FPG_A-C.bit are files created by the Xilinx program bitgen, part of the ISE 7.1 tools. The file main.txt contains instructions for the DN8000K10PCI configuration circuitry, including which FPGAs to configure, and to which frequency the global clock networks should be automatically adjusted.

2. Insert the SmartMedia card labeled "Reference Design" into the DN8000K10PCI's SmartMedia slot, contacts down.

## 3.5 Connect cables

The configuration circuitry can accept user input to control FPGA configuration or provide feedback during the configuration process. The configuration circuitry IO can also be used to transfer data to and from the user design.

1. Use the provided ribbon cable to connect the MCU RS232 port (P2) to a computer serial port to view feedback from the configuration circuitry during FPGA configuration. Run a serial terminal program on your PC (On Windows you can use HyperTerminal Start->Programs->Accessories->Communications->HyperTerminal) and make sure the computer serial port is configured with the following options:

   - Bits per second: 19200
   - Data bits: 8
   - Parity: None
   - Stop Bits: 1
   - Flow control: None
   - Terminal Emulation: VT100

2. Use the provided USB cable to connect the DN8000K10PCI to a Windows computer (Windows XP is recommended).

3. Plug an ATX power supply into J1, or plug the DN8000K10PCI into a PCI slot. **Do not plug an external power supply into J1 if the DN8000K10PCI is plugged into a PCI slot**. Turn on the ATX power supply.

When the DN8000K10PCI powers on, it automatically loads Xilinx FPGA design files (ending with a .bit extension), found on the SmartMedia card in the SmartMedia slot into the FPGAs.

## 3.6 View configuration feedback over RS232

As the DN8000K10PCI powers on, your RS232 terminal (connected to P2) will display useful information about the Configuration process.

### 3.6.1 Watch the configuration status output

| | |
|---|---|
| No USB cable detected, rebooting from FLASH...please wait | |
| Setting ACLK…<br>N 01 M: 000001000<br>DONE<br>Setting BCLK...<br>N: 01 M: 000001000<br>DONE<br>Setting DCLK...<br>N: 01 M: 000001000<br>DONE<br>Setting R1CLK...<br>N: 01 M: 000001000<br>DONE<br>Setting R2CLK...<br>N: 01 M: 000001000<br>DONE | The global clocks (ACLK, BCLK, DCLK) are frequency-configurable. The M binary sequence represents the multiplication applied to the installed crystal. The N represents the division applied. U6, U14, U20, U31 and the ICS8442AY datasheet.<br>The MCU is setting the clocks to their default values ACLK 200Mhz, BCLK 108.8Mhz, DCLK 128Mhz, R1CLK (not available on DN8000K10PCI), R2CLK (**DEFAULT**) |
| =-=- DN8000k10PCI MCU FLASH BOOT -=-=<br>-- FPGAS STUFFED --<br>A B | The MCU detects which FPGAs are present |
| -- SMART MEDIA INFO --<br>MAKER ID: EC<br>DEVICE ID: 75<br>SIZE: 32 MB | The MCU detects if a SmartMedia card is present |
| -- FILES FOUND ON SMART MEDIA CARD<br>FPGA_B.BIT<br>FPGA_A.BIT<br>MAIN~1.TXT<br>MAIN.TXT<br>-- CONFIGURATION FILES --<br>FPGA A: FPGA_A.BIT<br>FPGA B: FPGA_B.BIT | The MCU tries to access the SmartMedia card. If the MCU is not successful in reading the files on the SmartMedia card, be sure you have not formatted the card in Windows. Windows uses a non-standard format for media cards and will make the card unreadable. You can download a format utility from dinigroup.com to repair your incorrectly-formatted SM card. |
| --OPTIONS--<br>Message level set to default: 2<br>Sanity check is set to default: ON<br>N: 00 M: 000001010<br>DONE<br>Setting BCLK...<br>N: 01 M: 000001100<br>DONE<br>Setting DCLK...<br>N: 01 M: 000001000<br>DONE<br>Setting R1CLK...<br>N: 01 M: 000001000<br>DONE<br>Setting R2CLK...<br>N: 01 M: 000001000<br>DONE | The MCU reads the contents of the file MAIN.TXT and executes each instruction line.<br>Here the MCU is setting the clocks according to instructions in MAIN.TXT |
| **********CONFIGURING FPGA: A**********<br>-- Performing Sanity Check on Bit File --<br>-- BIT FILE ATTRIBUTES --<br>FILE NAME: FPGA_A.BIT<br>FILE SIZE: 003A943B bytes<br>PART: 4vlx100ff151317:09:38<br>DATA: 2005/07/25<br>TIME: 17:09:38 | The MCU is configuring FPGA A according to instructions in MAIN.TXT<br>The sanity check option reads the design (".bit") file headers and verifies that the design is compiled for the same type of FPGA that the MCU detects on your DN8000K10PCI. If the design and FPGA do not match, the MCU will reject the file and flash the Error LED. You may need to disable to sanity check option (See Chapter X, section X) if you want to encrypt or compress your configuration |

| | |
|---|---|
| Sanity check passed<br>.....<br>.....<br>.....DONE WITH CONFIGURATION OF FPGA: A | files. |
| **********CONFIGURING FPGA B**********<br>-- Performing Sanity Check on Bit File --<br>-- BIT FILE ATTRIBUTES --<br>FILE NAME: FPGA_B.BIT<br>FILE SIZE: 003A943B bytes<br>PART: 4vlx100ff151317:05:01<br>DATA: 2005/07/19<br>TIME: 17:05:01<br>Sanity check passed<br>.....<br>.....<br>.....DONE WITH CONFIGURATION OF FPGA: B | The MCU is configuring FPGA B according to instructions in MAIN.TXT<br><br>The MCU is setting the temperature threshold to cause a board reset. |
| -- TEMPERATURE SENSORS --<br>A YES<br>B YES<br>FPGA Temperature Alarm Threshold: 80 degrees C<br><br>DN8000k10PCI MAIN MENU (Jul 27 2005 10:38:05)<br>1.) Configure FPGAs using "MAIN TXT"<br>2.) Interactive configuration menu<br>3.) Check configuration status<br>4.) Change MAIN configuration file<br>5.) List files on Smart Media<br>6.) Display Smart Media text file<br>7.) Change RS232 PPC Port<br><br>8.) Set FPGA Address<br>9.) Write to FPGA at current address<br><br>a.) Read from FPGA at current address<br>g.) Display FPGA Temperatures<br>h.) Set FPGA Temperature Alarm Threshold<br>ENTER SELECTION: | Here is the MCU main menu<br><br>Options 8,9, and A are only available when the DN8000K10PCI reference design is loaded. For more information on how the MCU communicates with the reference design, see Chapter X, The Reference Design. |

Figure 5 RS232 Output

You should see the DN8000K10PCI MCU main menu. If the reference design is loaded in the Virtex 4 FPGAs, then you should see the above on your terminal. Try pressing 3 to see if the configuration circuit was successful in programming the FPGAs.

```
ENTER SELECTION: 3

******************** CONFIGURATION STATUS ******************

FPGA B NOT configured
```

The easiest way to verify your FPGAs are configured is to look at DS18, DS14, DS16 located above each FPGA. When the green LEDs are lit, the FPGA under it is successfully configured.

### 3.6.2 Interactive configuration

If you want to put multiple designs on a single Smart Media card, you can use the interactive configuration menu to select which .bit file to use on each FPGAs. Select menu option 2.

```
ENTER SELECTION: 2

-=-= INTERACTIVE CONFIGURATION MENU =-=-


1) Select bit files to configure FPGA(s)

2) Set verbose level (current level = /)

3) Enable sanity check for bit files

M) Main Menu


Enter Selection:
```

Figure 6 Interactive Config Menu

### 3.6.3 Read temperature sensors

The DN8000K10PCI is equipped with temperature sensors to measure and monitor the temperature on the die of the Virtex 4 FPGAs. According to the Virtex 4 datasheet, the maximum recommended operating temperature of the die is 85C degrees. If the microcontroller measures a temperature above 80 degrees, it will reset the DN8000K10PCI.

If you think your DN8000K10PCI is resetting due to temperature overload, you can use the temperature monitor menu to measure the current junction temperature of each FPGA.

```
ENTER SELECTION: g


-- FPGA TEMPERATURES (Degrees Celsius [+/- 4]) --

B 29

-- Set FPGA Temperature Alarm Threshold  --

(degrees C, decimal values, range [1-127])

Old Threshold: 80

New Threshold: 85

Threshold Updated: 85 Degrees C
```

Figure 7 Temperature Threshold Menu

The Virtex 4 FPGA can operate as hot as 120C degrees before damaging the part, although timing specifications are not guaranteed. The MCU allows you to change the reset threshold,

although we recommend improving your heat dissipation to maintain a low junction temperature.

### 3.6.4 Multiplex Serial port

The DN8000K10PCI has one serial port (P1) for user use. This single port is multiplexed so that any FPGA can access it through its RX and TX signals. You can use the RS232 MCU interface to change the FPGA to which P1 is connected.

```
ENTER SELECTION: 7


PORT 1: D
 PORT 2: A
 PORT 3: A
 PORT 4: A


Enter Port to change (1-4, q to quit): 1
Enter FPGA to set port to (A-I): B
Do you want to change more RS232 Ports (y or n)?: n
```

Figure 8 RS232 Port Menu

The DN8000K10PCI only has one serial port (Port 1). Changing ports 2-4 will have no effect.

## 3.7 Check LED status lights

The DN8000K10PCI has many status LEDs to help the user confirm the status of the configuration process.

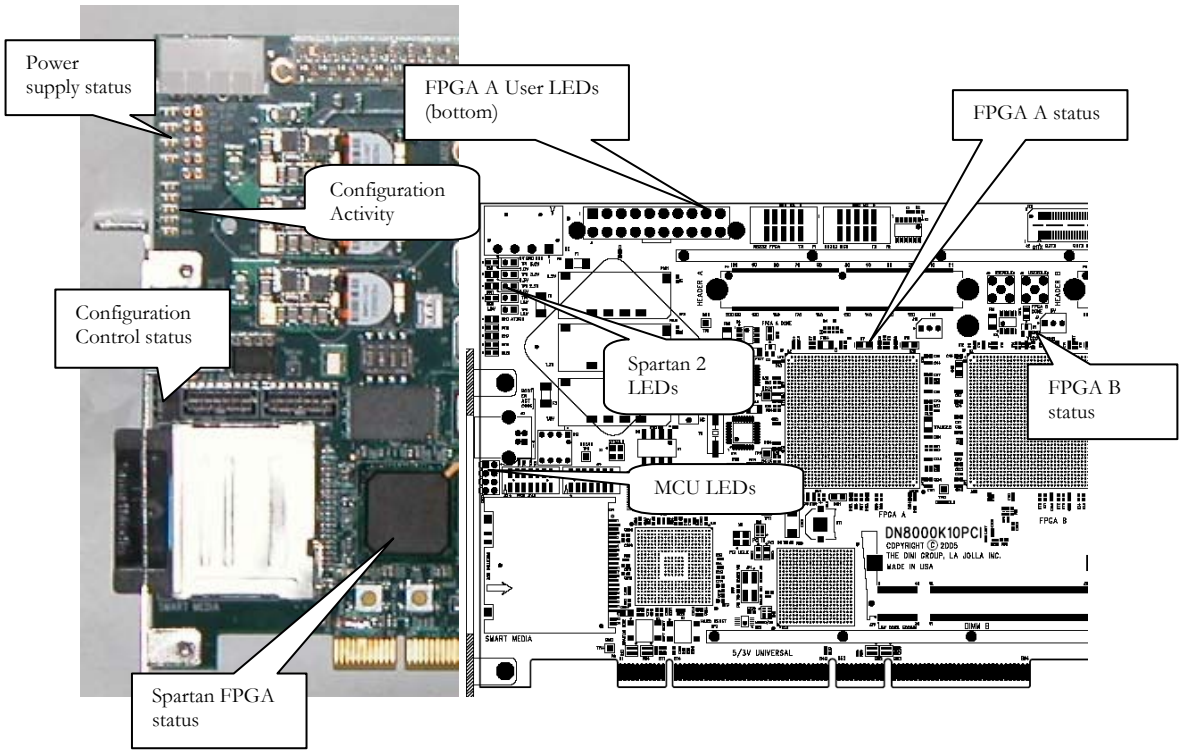



Figure 9 Configuration Status LEDs

1. Check the power voltage indication LEDs to confirm that all voltage rails of the DN8000K10PCI are present. From the top, the LEDs indicate the presence of 5V, 3.3V, 2.5V, and "ATX POWER OK" Green lit LED's on the voltage present LEDs indicate the rails are greater than 1.7V. A green lit "ATX power OK" indicates that the voltage monitors inside the ATX power supply are within acceptable operating ranges (5V is 4.5 – 5.5V, 3.3V is 3.0-3.6V). If this LED is not lit green, the DN8000K10PCI might not function properly.

2. Check the Configuration status LEDs. These LEDs are visible from outside the case when the DN8000K10PCI is installed in an ATX case. Under error conditions, all four red LEDs will blink.

3. Check the Spartan FPGA status LED, DS24. This LED indicates that the Spartan II FPGA has been configured. If this LED is not lit soon after power on, then there may be a problem with the firmware on the DN8000K10PCI. This LED off or blinking may indicate a problem with one of the board's power supplies.

4. Check the FPGA A status LED, DS18 to the upper left of FPGA A. This green LED is lit when FPGA A is configured and operational. This light should be on if you loaded the reference design from the SmartMedia card.

5. Check the FPGA B status LED, DS14 directly above FPGA B. This light should be lit green if your DN8000K10PCI was installed with the FPGA B option, and the reference design is loaded.

6. Check the FPGA C status LED, DS16 to the upper left of FPGA C. This green LED will light if you have the FPGA C option and the FPGA is configured.

7. Check the FPGA A User LEDs on the bottom side of the DN8000K10PCI. If you have successfully loaded the Dini Group's DN8000K10PCI reference design, these should be lit in the pattern "00011010".

8. Check the FPGA C User LEDs on the bottom side of the DN8000K10PCI. If you have ordered the "FX" FPGA C option, and the reference design is loaded, these be lit in the pattern "1110110111001011".

9. If you suspect one or more FPGAs did not configure properly, check the configuration circuitry's status lights. These are four right-angle mounted LEDs viewable out the side of the PC case. If there has been an error, three of the LEDs will blink. If there has been no error, there should be two LEDs ON and two OFF. If there was an error, the easiest way to determine the cause of the error is to connect a terminal to the RS232 port (P2) and try to configure again. Configuration feedback will be presented over this port.

You should also notice the Fans mounted above the 3 Virtex 4 FPGAs and the Fan mounted above the power supplies spinning.

| **Assembly Number** | **Signal** | **Comment** |
|---|---|---|
| DS9 | 5.0V_PRESENT | The 5.0V power rail is present (above ~1.7V) |
| DS10 | 3.3V_PRESENT | The 3.3V power rail is present (above ~1.7V) |
| DS11 | 2.5V_PRESENT | The 2.5V power rail is present (above ~1.7V) |
| DS12 | 1.8V_PRESENT | The 1.8V power rail is present (above ~1.7V) |
| DS13 | ATX_POK | The ATX power supply is generating 5.0V and 3.3V |

| | | |
|---|---|---|
| | | within 5% at the source |
| DS15 | SPARTAN_LED3 | This LED will flicker when there is PCI configuration activity (Bar 0) |
| DS17 | SPARTAN_LED2 | This LED will flicker when there is Main Bus activity (See section X.X.X) |
| DS19 | SPARTAN_LED1 | This LED will flicker when there is USB activity (Bulk Transfer) |
| DS20 | SPARTAN_LED0 | This LED will flicker when there is SmartMedia card activity. |
| DS21.1 | MCU_LED0 | MCU_LED[1:0] Codes: |
| DS21.2 | MCU_LED1 | 01 FPGA A is configuring<br><br>10 FPGA B is configuring<br><br>11 FPGA C is configuring |
| DS21.3 | MCU_LED2 | The last FPGA configuration was successful |
| DS21.4 | MCU_LED3 | Blinking: There was a configuration error. Use the RS232 port to read the error. Off: No error. On: The last configuration command was successful |
| DS24 | SPARTAN_DONE | The Spartan 2 configuration FPGA is configured. This light will turn off if the board is in power reset |

| | | |
|---|---|---|
| DS22 | PCI_IN | PCI data is arriving from the PCI bus |
| DS23 | PCI_OUT | PCI data is being transmitted onto the PCI bus |
| DS18 | FPGA_A_DONE | The Virtex 4 FPGA A is configured |
| DS14 | FPGA_B_DONE | The Virtex 4 FPGA B is configured |
| DS16 | FPGA_C_DONE | The Virtex 4 FPGA C is configured |
| DS8 | SFP2_LOS | SFP module 2 Loss-of-signal |
| DS4 | SFP2_FAULT | SFP module 2 transmitter fault |
| DS5 | XFP2_INT | XFP module 2 error |
| DS1 | XFP2_FAULT | XFP module 2 transmitter fault |
| DS6 | SFP1_LOS | SFP module 1 Loss-of-signal |
| DS2 | SFP1_FAULT | SFP module 1 transmitter fault |
| DS7 | XFP1_INT | XFP module 1 error |
| DS3 | XFP1_LOS | XFP module 1 Loss-of-signal |
| DS48, DS47, DS46, DS45, DS44, DS43, DS42, DS41, | | User LEDs from FPGA C |

| | | |
|---|---|---|
| DS40,DS39, DS38, DS37, DS36, DS35, DS34, DS33, | | User LEDs from FPGA C |
| DS32, DS31, DS30, DS29, DS28, DS27, DS26, DS25 | | User LEDs from FPGA A |

Figure 10 DN8000K10PCI LEDs

# 4 Using the Reference Design with the Provided Software

To communicate with the reference design (or user design) on the emulation board, the DN8000K10PCI provides three options out of the box.

- USB
- PCI
- RS232

The USB and PCI interfaces allow configuration of the FPGAs and bulk data transfer to and from the User design. The RS232 interface allows low-speed data transfers to and from the User design, and control and monitoring of the configuration process.

This section will get you started and show you how to operate the provided software. For detailed information about the reference design and implementation details, see Chapter 5, The Reference Design.

## 4.1 Operating the USB controller program

Use the provided USB monitoring software to verify that the design is loaded into the FPGAs.

1. Insert the CDROM that came with your DN8000K10PCI into the CDROM drive of your computer.

2. Connect the USB cable to your DN8000K10PCI and a Windows XP PC. (Before or after the DN8000K10PCI has powered on)

3. When you connect the USB cable to your PC for the first time, Windows detects the DN8000K10PCI and asks for a driver. The board should identify itself as a "DiNi Prod FLASH BOOT". When the new device detected window appears, select the option "install from a list" -> select "search for the best driver in these locations". Select "include the location in the search" and browse to the product CD in "Source Code\AETEST_USB\driver\win_wdm\" ->select "finish"

4. After Windows installs the driver, you will be able to see the following device in the USB section of Windows device manager: "DiniGroup DN8000K10PCI FLASH boot".

5. Run the USB controller application found on the product CD in "Source Code\USBController\USBController.exe".

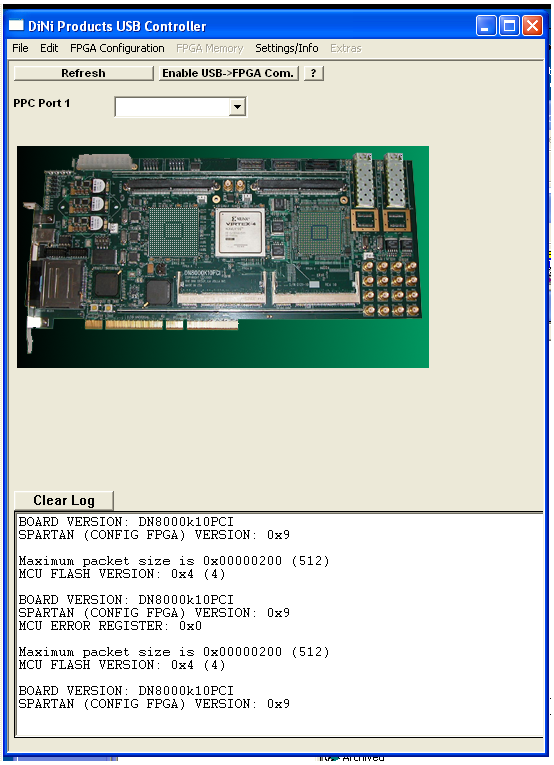

DiNi Products USB Controller
File Edit FPGA Configuration FPGA Memory Settings/Info Extras
Refresh
Enable USB->FPGA Com.
?
PPC Port 1
Clear Log
BOARD VERSION: DN8000k10PCI
SPARTAN (CONFIG FPGA) VERSION: 0x9
Maximum packet size is 0x00000200 (512)
MCU FLASH VERSION: 0x4 (4)
BOARD VERSION: DN8000k10PCI
SPARTAN (CONFIG FPGA) VERSION: 0x9
MCU ERROR REGISTER: 0x0
Maximum packet size is 0x00000200 (512)
MCU FLASH VERSION: 0x4 (4)
BOARD VERSION: DN8000k10PCI
SPARTAN (CONFIG FPGA) VERSION: 0x9

Figure 11 USB Controller Window

6. This window will appear showing the current state of the DN8000K10PCI. Next to each FPGA a green light will appear if that FPGA is configured successfully. The above window shows the USB Controller connected to a DN8000K10PCI with a single FPGA in the B position. If you have the reference design loaded and a DDR2 SODIMM installed, you can use the USB Controller to run tests of the SODIMM. From the FPGA Memory menu, select Test DDR.

7. Clear the FPGAs of their configurations. Right-click on an FPGA and select from the popup menu, "Clear FPGA". The green light above the FPGA on the GUI and on the board should stop shinning green.

8. Configure an FPGA using the USB Controller program. Right-click on an FPGA and select Configure FPGA via USB from the popup menu. The program will open a dialog

box for you to select the configuration file to use for configuration. Browse to the provided user's CD D:\FPGA_Reference_Designs\Programming_Files\DN8000K10PCI\MainTest\LX100 If you are configuring an LX200 or FX60 devices you should select a bit file from the LX200 or FX60 directories instead. If you are configuring FPGA B or FPGA C, you should select fpga_b.bit or fpga_c.bit instead.

```
Done
FPGA B cleared successfully.
FPGA A cleared successfully.
Doing a sanity check...Sanity Check passed. Configuring FPGA
B via USB...please wait.
File
D:\\dn_BitFiles\DN8000K10PCI\MainTest\LX100\fpga_b.bit
transferred.
Configured FPGA B via USB
```

Figure 12 USB Controller Log Output

9. The message box below the DN8000K10PCI graphic should display some information about the configuration process

The USB Controller program also allows you to easily configure and transfer data to and from the user design on the emulation board. More information is provided in Chapter 3, "Controller Software"

## 4.2 Communicating to the User Design over the Serial Port

You may want to communicate with your design over the user serial port (P1). Only one FPGA can use P1 at a time. Before you can communicate to your design, change the RS232 multiplexing settings as described in Section 3.6.4. You can also change the RS232 multiplexing settings using the USB Controller software.

Connect a second RS232 cable to P1, the FPGA RS232. It is located right next to the configuration RS232 port, P2. If you have the reference design loaded, the FPGA RS232 port runs at 19200 bps, 8 bit, no parity. By default, the FPGA RS232 port is connected to FPGA A. One the computer's terminal, the reference design is programmed to digitally loop back the input to the output. If on the terminal you can read your own output, then the reference design was able to capture the RS232 signal and generate an RS232 signal that your computer could capture.

If you are familiar with previous Dini Group products, the reference design test outputs could be read from this serial port. On the DN8000K10PCI, you must use the AETEST application to read the results of self-test.

## 4.3 Using AETEST to run hardware tests

AETest is the program that you can use to verify the hardware on the DN8000K10PCI, as well as to demonstrate the reference design function. The following instructions assume you have a PC running the Windows XP operating system. The user CD includes a Windows version of the AETest program. If you plan to use the DN8000K10PCI in stand-alone mode, connect the DN8000K10PCI to your WindowsXP computer and use aetest_usb in

D:\USB_Software_Applications\USB_CMD_Line_AETEST_USB. If the computer asks for a driver, click "Have Disk" and browse to
D:\USB_Software_Applications\driver\windows_wdm\dndevusb.inf
If you are going to use the DN8000K10PCI in a PCI slot, turn off the computer, insert the DN8000K10PCI into an unused PCI slot, and turn the computer on. If the operating system asks for a driver, click "Have Disk" and browse to

Then run the PCI version of the AETest application at
D:\USB_Software_Applications\USB_CMD_Line_AETEST_USB

### 4.3.1 AETest, Windows98 version

A win98 driver is provided for using the DN8000K10PCI over USB, but support for this driver is limited.

### 4.3.2 AETest, DOS version

<This section is out of date. Contact support@dinigroup.com>

The AETest application will also run under DOS. The DOS version of AETest will not run under Windows's DOS emulation mode. You must boot into DOS using a boot disk.

1) Create a windows boot floppy disk. The easiest way to do this is to format a disk using WindowsXP with the "Create an MS-DOS startup dick" option checked.

2) D: \AETest\aetest_floppy
contains some files you need to add to the boot disk. Copy the contents of this directory to the disk. Add the program
D:\AETest\DJGPP\CWSDPMI.EXE
to the floppy. The DOS version of AETest requires CWSDPMI.EXE to access the PCI bus.

3) Plug DN8000K10PCI into the PCI slot.

4) Boot from the floppy

5) Run AETESTDJ.EXE

### 4.3.3 AETest on Linux or Solaris

To use the AETest application on Linux or Solaris, you must compile the source code included on the User CD.

### 4.3.4 Use AETest

The AETest application should display it's main menu.

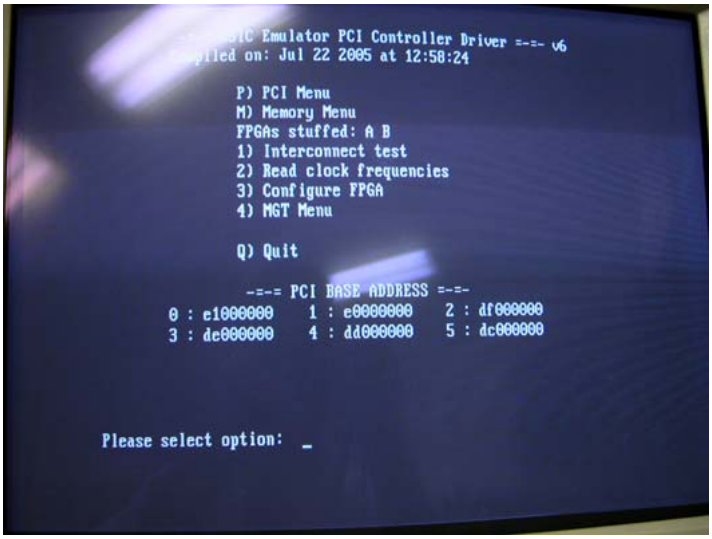

Emulator PCI Controller Driver =-=- v6
Compiled on: Jul 22 2005 at 12:58:24
P) PCI Menu
M) Memory Menu
FPGAs stuffed: A B
1) Interconnect test
2) Read clock frequencies
3) Configure FPGA
4) MGT Menu
Q) Quit
-=-= PCI BASE ADDRESS =-=-
0 : e1000000 1 : e0000000 2 : df000000
3 : de000000 4 : dd000000 5 : dc000000
Please select option: _

Figure 13 AETEST Main Menu

Run one of the tests. Choose option 1. Remember, the FPGA you test has to be loaded with the reference design, or the test will fail.

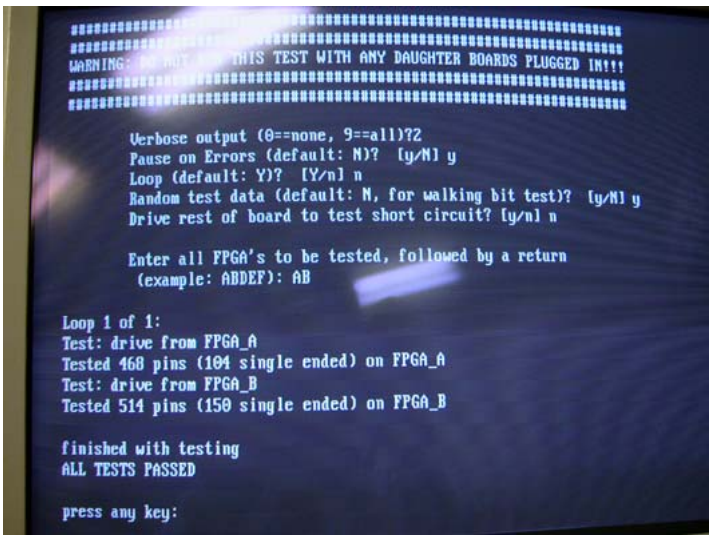

THIS TEST WITH ANY DAUGHTER BOARDS PLUGGED IN!!!
Verbose output (0==none, 9==all)?2
Pause on Errors (default: N)? [y/N] y
Loop (default: Y)? [Y/n] n
Random test data (default: N, for walking bit test)? [y/N] y
Drive rest of board to test short circuit? [y/n] n
Enter all FPGA's to be tested, followed by a return
(example: ABDEF): AB
Loop 1 of 1:
Test: drive from FPGA_A
Tested 168 pins (104 single ended) on FPGA_A
Test: drive from FPGA_B
Tested 514 pins (150 single ended) on FPGA_B
finished with testing
ALL TESTS PASSED
press any key:

Figure 14 AETest Interconnect Menu

For more information on the AETEST program, see Chapter 3.

## 4.4 Moving On

Congratulations! You have just programmed the DN8000K10PCI and learned all of the features that you must know to start your emulation project. If you are new to Xilinx FPGA, you might want move to chapter 4, introduction to ISE and Virtex 4 and start adding your Verilog code to the reference design. The User CD contains a netlist of the board and example UCF files. (From the reference design) All of the source code for the reference design in Verilog, including embedded PowerPC code and utility is included on the provided CD.

Chapter

3

# Controller Software

## 1 USB Controller

USBController application is used to communicate with the DN8000K10PCI.

All USBController source code is included on the CD-ROM shipped with the DN8000K10PCI. The USBController can be installed on Windows 98/ME/2000/XP. There is a command line version called AETEST_USB that can be installed on Linux and Solaris.

The USBController Application contains the following functionality:
- Verify Configuration Status
- Configure FPGA(s) over USB
- Configure FPGAs via SmartMedia card
- Clear FPGA(s)
- Reset FPGA(s)
- Set Global clocks frequency
- Set RocketIO CLK Frequency
- Update MCU FLASH firmware

The following function interface with the Dini Group reference design.
- Read/Write to FPGA(s) – An Address map is contained in the Reference Design chapter
- Test DDRs/FLASH/Reigsters/FPGA Interconnect

### 1.1 Menu Options

#### 1.1.1 File Menu

The File Menu has the following 2 options:

a. Open – opens a file with the selected text editor (notepad by default). To change the text editor see Settings/Info Menu section

b. Exit – Closes the USBController application

### 1.1.2 Edit Menu

The Edit Menu performs the basic edit commands on the command log in the bottom half of the USBController window.

### 1.1.3 FPGA Configuration Menu

The FPGA Configuration Menu has the following options:

(1) Configure via USB (individually) – After selecting this option a window will pop and ask which FPGA you want to configure and then what bit file you want to configure the selected FPGA with. The status of the FPGA configuration will detailed in the log window and the DN8000K10PCI will be updated after the bit file has been transferred.

(2) Configure via USB using file – This option allows the user to configure more than one FPGA over USB at a time. To use this option you must create a setup file that contains information on which FPGA(s) should be configured and what bit files should be used for each FPGA. The file should be in the following format, the first character of each line represents which FPGA you want configured (a-f or A-F), this letter should be followed by a colon and then the path to the bit file to use for this FPGA. The path to the bit file is relative to the directory where this setup file is, or you can use the full path. Below is an example of an accepted setup file:

A: fpga_a.bit

B: fpga_b.bit

C: fpga_c.bit

(3) Configure via SmartMedia Card – This option allows the user to use a SmartMedia card to configure the FPGAs. Please section Creating Configuration File "main.txt" for information on what files should be on the SmartMedia card to use this option.

(4) Clear All FPGAs – This option will deconfigure all FPGAs.

(5) Reset – This options sends an active low reset (active for approx. 20ns) to all FPGAs on the signal called RESET_FPGASn which is connected to the following I/O pins:

FPGA A: AK19

FPGA B: K21

FPGA C: AG18

### 1.1.4 Settings/Info Menu

The Settings/Info Menu has the following options

(1) Set FPGA RocketIO CLK Frequency – When the DN8000K10PCI is first powered up the RocketIO CLK inputs to the FPGAs are inactive. The RocketIO CLK Inputs are connected to the following FPGA Differential CLK inputs on all FPGAs: F21/G21 and AT21/AU21. This menu option allows the user to specify what frequency the RocketIO CLKs should be set at for each FPGA. The supported frequency range is 31.25MHz – 700MHz. After selecting this option, a pop-up window will ask which FPGA's RocketIO Frequency you want to set (or you can choose to set all to the same frequency), and then what frequency you want. Check the log window to verify what frequency the CLKs were actually set at.

(2) Set Global clock frequencies

The clocks on the DN8000K10PCI are automatically adjusted to the user's desired frequency by reading the setup file on the SmartMedia card. If you wish to change the frequency after power-on, or do not want to use a SmartMedia card, you can set the frequency in the USB program.

ACLK) ACLK is generated from a 25MHz crystal. Available frequencies are:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 31.25 | 34.375 | 37.5 | 40.625 | 43.75 | 46.875 | 50 | 53.125 | 56.25 | |
| 59.375 | 62.5 | 65.625 | 68.75 | 71.875 | 75 | 78.125 | 81.25 | 84.375 | 87.5 |
| 93.75 | 100 | 106.25 | 112.5 | 118.75 | 125 | 131.25 | 137.5 | 143.75 | 150 |
| 156.25 | 162.5 | 168.75 | 175 | 187.5 | 200 | 212.5 | 225 | 237.5 | 250 |
| 262.5 | 275 | 287.5 | 300 | 312.5 | 325 | 337.5 | 350 | 375 | 400 |
| 425 | 450 | 475 | 500 | 525 | 550 | 575 | 600 | 625 | 650 |
| 675 | 700 | | | | | | | | |

BCLK) BCLK is generated from a 14.318 Mhz crystal. Supported frequencies are:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 32.22 | 34.01 | 35.80 | 37.58 | 39.37 | 41.16 | 42.95 | 44.74 | 46.53 | 48.32 |
| 50.11 | 51.90 | 53.69 | 55.48 | 57.27 | 59.06 | 60.85 | 62.64 | 64.43 | 66.22 |
| 68.01 | 69.80 | 71.59 | 73.38 | 75.17 | 76.96 | 78.75 | 80.54 | 82.33 | 84.12 |
| 85.91 | 89.49 | 93.07 | 96.65 | 100.2 | 103.8 | 107.4 | 111.0 | 114.5 | 118.1 |
| 121.7 | 125.3 | 128.9 | 132.4 | 136.0 | 139.6 | 143.2 | 146.8 | 150.3 | 153.9 |
| 157.5 | 161.1 | 164.7 | 168.2 | 171.8 | 179.0 | 186.1 | 193.3 | 200.5 | 207.6 |
| 214.8 | 221.9 | 229.1 | 236.2 | 243.4 | 250.6 | 257.7 | 264.9 | 272.0 | 279.2 |
| 286.4 | 293.5 | 300.7 | 307.8 | 315.0 | 322.2 | 329.3 | 336.5 | 343.6 | 358.0 |
| 372.3 | 386.6 | 400.9 | 415.2 | 429.5 | 443.9 | 458.2 | 472.5 | 486.8 | 501.1 |
| 515.4 | 529.8 | 544.1 | 558.4 | 572.7 | 587.0 | 601.4 | 615.7 | 630.0 | 644.3 |
| 658.6 | 672.9 | 687.3 | | | | | | | |

DCLK) DCLK is generated from a 16.0 Fundamental crystal. Supported frequencies:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 32 | 34 | 36 | 38 | 40 | 42 | 44 | 46 | 48 | 50 |
| 52 | 54 | 56 | 58 | 60 | 62 | 64 | 66 | 68 | 70 |
| 72 | 74 | 76 | 78 | 80 | 82 | 84 | 86 | 88 | 92 |
| 96 | 100 | 104 | 108 | 112 | 116 | 120 | 124 | 128 | 132 |
| 136 | 140 | 144 | 148 | 152 | 156 | 160 | 164 | 168 | 172 |
| 176 | 184 | 192 | 200 | 208 | 216 | 224 | 232 | 240 | 248 |
| 256 | 264 | 272 | 280 | 288 | 296 | 304 | 312 | 320 | 328 |
| 336 | 336 | 344 | 352 | 368 | 384 | 400 | 416 | 432 | 448 |
| 464 | 480 | 496 | 512 | 528 | 544 | 560 | 576 | 592 | 608 |
| 624 | 640 | 656 | 672 | 688 | | | | | |

(3) Change Text Editor – This options allows the user to select a text editor to use (the default editor is notepad).

(4) FPGA Stuffing Information – This option will display the type of FPGAs that are stuffed on the DN8000K10PCI.

(5) MCU Firmware Version – This option will display the MCU Firmware version in the log window.

(6) BOARD/SPARTAN Version – This option will display the Board Version along with the Spartan (Config Fpga) Version.

### 1.1.5 Configuration Register Map

The DN8000K10PCI firmware is updated constantly to add compatibility for new products and add features. The information in this section may change after this manual is printed. The memory space of the MCU is 16 bits wide.

This table describes registers within the Configuration FPGA that are accessible from the memory space of the MCU.

| REGISTER | ADDRESS | FUNCTION |
|---|---|---|
| BITS_1 | DF07 | BIT7: mcu_fpga_config_rd<br>BIT6: mcu_fpga_config_done<br>BIT5: FPGA_ack<br>BIT4: r_FPGA_PROGn,<br>BIT3-2: mcu_mode<br>BIT1: mcu_sm_rdy<br>BIT0: mcu_reading |
| BITS_2 | DF08 | BIT4: FPGA_DONE<br>BIT3 CPLD_idle<br>BIT2: SM_RDYBUSYn<br>BIT1: FPGA_INITn<br>BIT0: mcu_encrypt |

| | | |
|---|---|---|
| SM_SIGNALS | DF09 | |
| MCU_XADDR | DF0A | address register for upper FLASH/SRAM bits |
| MCU_CNTL | DF0B | address register for upper FLASH/SRAM bits |
| FPGA_SELECT | DF0C | FPGA_select[3:0] = bits 3:0<br>Selects which FPGA PPC is connected to<br>PPC Port 1 and 2:<br>FPGAs are represented by 4 bits:<br>A = 000, B = 0x1, ..., I = 0x8<br>Port 1 is lower 4 bits, |
| PPC_RS232_12SELECT | DF0D | Port 2 is upper 4 bits |
| PPC_RS232_34SELECT | DF0E | Select which FPGA PPC(s) is connected to PPC<br>Port 3 and 4 |
| FPGA_CNTRL | DF0F | bits[1:0] = 01 (write address), 10 (data write),<br>11 (data read)<br><br>bits[4:2] = r_fpga_state<br>bit5 = r_fpga_auto_inc<br>(when high, auto increments fpga_addr after access)<br>bit6 = r_fpga_rd_done |
| FPGA_BE | DF10 | select byte in addr, read, and data bytes |
| FPGA_RD_DATA | DF11 | |
| FPGA_WR_DATA | DF12 | |
| FPGA_ADDR | DF13 | |
| FPGA_ERROR | DF14 | |
| GPIF_DATA | DF20 | |
| GPIF_ERROR | DF21 | |
| HOLD_DONES | DF22 | |
| STATES | DF2 | [7:4] = GPIF_STATE, [3:0] = FPGA_STATE |
| FPGA_FREQ_H | DF24 | |
| FPGA_FREQ_SEL | DF25 | |
| FPGA_FREQ_L | DF26 | |
| MCU_STUFFING1 | DF27 | |
| MCU_STUFFING2 | DF28 | |
| ACLK_N_VAL | DF29 | Sets the divider value of GCLK0 |
| ACLK_M_VAL | DF30 | Sets the multiplier of GCLK0 |
| BCLK_N_VAL | DF31 | Sets the divider value of GCLK1 |
| BCLK_M_VAL | DF32 | Sets the multiplier of GCLK1 |
| DCLK_N_VAL | DF33 | Sets the divider value of GCLK2 |
| DCLK_M_VAL | DF34 | Sets the multiplier of GCLK2 |
| BOARD_VERSION_DUP | DF46 | |
| CF_REG_OFFSET | DF6X | Various CompactFlash card controls. Main.txt is read through these registers. |
| FPGA_COMMUNICATION | DF39 | Disables Main Bus interface |
| PENDING_CLKS | DF40 | Causes reprogramming of onboard synthesizers |
| CHECKSUM | DF45 | A checksum of USB configuration data |
| TEMP_SENSOR_A | DF50 | Temperature of FPGA A |
| TEMP_SENSOR_B | DF51 | Temperature of FPGA A |

| | | |
|---|---|---|
| TEMP_SENSOR_C | DF52 | Temperature of FPGA A |
| SERIAL_NUM_ADDR | DFFA | |
| SPARTAN_MKS_VERSION_ADDR | DFFB | |
| SPARTAN_VERSION_ADDR | DFFD | |
| BOARD_VERSION_NEW | DFFE | |
| BOARD_VERSION | DFFF | |

# 2 PCI AETEST Application

AETEST utility program can test and verify the functionality of the DN8000K10PCI Logic Emulation board, and provide data transfer to and from the User design.

All AETEST source code is included on the CD-ROM shipped with your DN8000K10PCI Logic Emulation kit. AETEST can be installed on a variety of operating systems, including:

- DOS and Windows 95/98/ME using DPMI (DOS Protected Mode Interface)
- Windows 98/ME using a VxD driver
- Windows 2000/XP (Windows WDM)
- Windows NT
- Linux
- Solaris

## 2.1 Functionality

The AETEST utility program contains the following tests:

- PCI Test
- Memory Tests (SRAM & DDR)
- FLASH Test
- Daughter Card Test (with or without cables)
- BAR Memory Range Tests

AETEST also provides the user with the following abilities:

- Recognize the DN8000K10PCI
- Display Vendor and Device ID
- Set PCI Device and Function Number
- Display all configured PCI devices
- Various loops for PCI device-function and ID numbers
- Write and Read Configuration DWORD
- Write DWORD, Read DWORD and Write/Read DWORD (Same Address)
- BAR Memory Fill, Write and Display
- Configure/Save BAR's from/to a file

## 2.2 Running AETEST

## 2.3 Compiling AETEST

There are two versions of AETest, one that controls the DN8000K10PCI from a PCI bus, and the other that controls the DN8000K10PCI over a USB connection. The source for the PCI version is found on the User CD
D:\PCI_Software_Applications\Aetest\
The source for the USB version is found on the User CD
D:\USB_Software_Applications\USB_CMD_Line_AETEST_USB\

You will likely want to interface to your ASIC emulation project using PCI. You may want to start your controller software by modifying and recompiling AETEST.

### 2.3.1 Compiling AETest for DOS

The DOS version of AETest requires DJGPP. You can find it at
http://www.delorie.com/djgpp/
Follow the installation instructions for DJGPP. The download comes with an utility to set your environment variables correctly.
D:\PCI_Software_Applications\Aetest\
Contains the source code for AETest. Copy this directory to your hard drive. Open the file Makefile. This file must be edited to define which operating system you wish to target. Uncomment the line

#DESTOS = DOS_DJGPP
and the line
include Makefile.make

In a DOS shell, run make

### 2.3.2 Compiling AETest for Windows XP

AETest for Windows requires visual studio to compile.

Copy the directory on the User CD
D:\PCI_Software_Applications\Aetest\
to your local machine. Open the file "Makefile" and uncomment the lines
#DESTOS = WIN_WDM
and
include Makefile.make
Run make.

# 3 Updating the Firmware

Dini Group may release firmware bug fixes or added features to the DN8000K10PCI. If a firmware update is released you will need to follow these instructions for access to the new features.

There are two firmware files that Dini Group may release, the first is a Micro controller (MCU) software update that is stored in a flash memory. This update can be accomplished easily from within the USBController application.

The second update that may be required is a Spartan FGPA core update. The configuration data for the Spartan FPGA is contained in a Xilinx configuration PROM. This update can be accomplished with the Xilinx JTAG programming program, iMpact.

## 3.1 Updating the MCU (flash) firmware

To protect against accidental erasure, the MCU firmware cannot be updated unless the board is put in firmware update mode during power-on. Find Switch block 1 on the DN8000K10PCI.

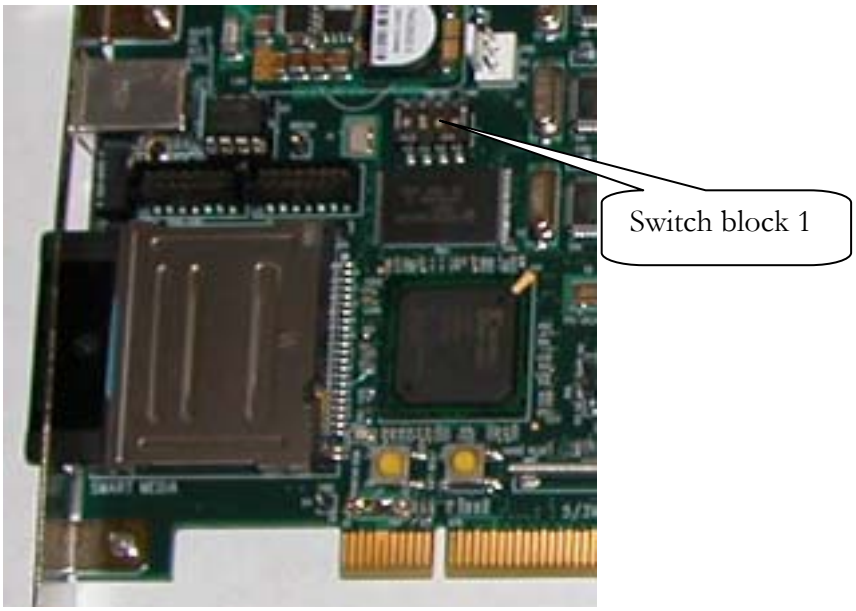



Figure 15 Switch block 1

Move switch S1 #3 to the ON position. Power on the DN8000K10PCI.

Open the USB Controller program. If the DN8000K10PCI powered on in firmware update mode, there will be an "Update Flash" button near the top of the USB Controller window. Click on this button.

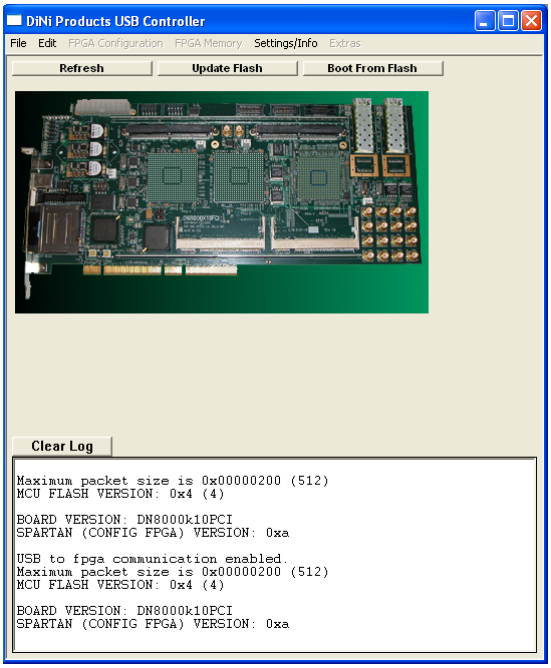

DiNi Products USB Controller
File Edit FPGA Configuration FPGA Memory Settings/Info Extras
Refresh
Update Flash
Boot From Flash
Clear Log
Maximum packet size is 0x00000200 (512)
MCU FLASH VERSION: 0x4 (4)
BOARD VERSION: DN8000k10PCI
SPARTAN (CONFIG FPGA) VERSION: 0xa
USB to fpga communication enabled.
Maximum packet size is 0x00000200 (512)
MCU FLASH VERSION: 0x4 (4)
BOARD VERSION: DN8000k10PCI
SPARTAN (CONFIG FPGA) VERSION: 0xa

Figure 16 USB Controller Firmware Update Mode

When the Open… dialog box appears, navigate to the Firmware image file supplied by Dini Group. The file name should be "flash_flp.hex". Press OK.

The USB Controller should freeze for about 10 seconds while the firmware update is taking place. When the download is complete, the Log window should print, "Update Complete"

Move Switch block 1 # 3 to the OFF position to put the DN8000K10PCI back into normal operation mode. Power cycle the board.

## 3.2 Updating the Spartan (EEPROM) firmware

Connect a Xilinx Parallel IV configuration cable to the parallel port of your computer. The Parallel IV cable requires external power to operate, so you may need to connect the keyboard connector power adapter. When the Parallel IV cable has power, the status LED on Parallel IV turns amber.

Use a 2mm IDC cable to connect the Parallel IV cable to the DN8000K10PCI connector J14.

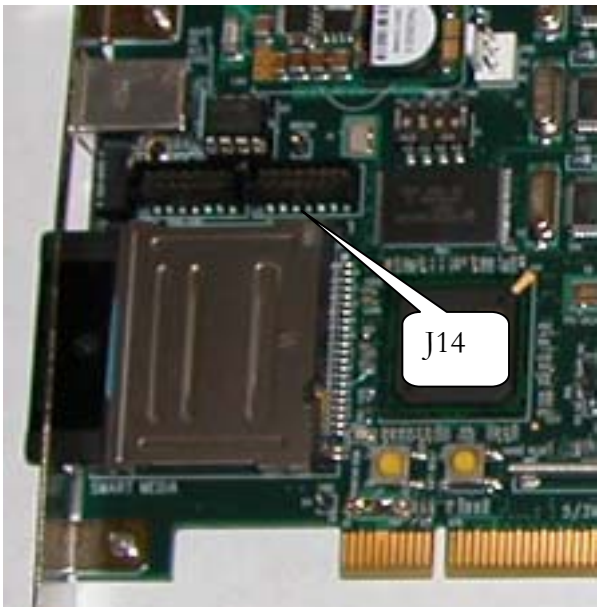



Figure 17 Firmware Update Header

Power on the DN8000K10PCI. When the Parallel IV cable is connected to a header, the status light turns green.

Open the Xilinx program Impact, usually found at Start->programs->Xilinx ISE 7.1->Accessories->impact

Impact may ask you to open an impact project. Hit cancel.

Choose the menu option File->Initialize Chain

Impact should detect 2 devices in the JTAG chain. Xc18v02 and Xc2s200. For each item in the chain Impact will direct you to select a programming file for each. For the xc18v02 device, select the Spartan Firmware update file provided by Dini Group. This file should be named prom.mcs. Hit Open. Impact will then ask for a programming file to program the xc2s200. Press Bypass.

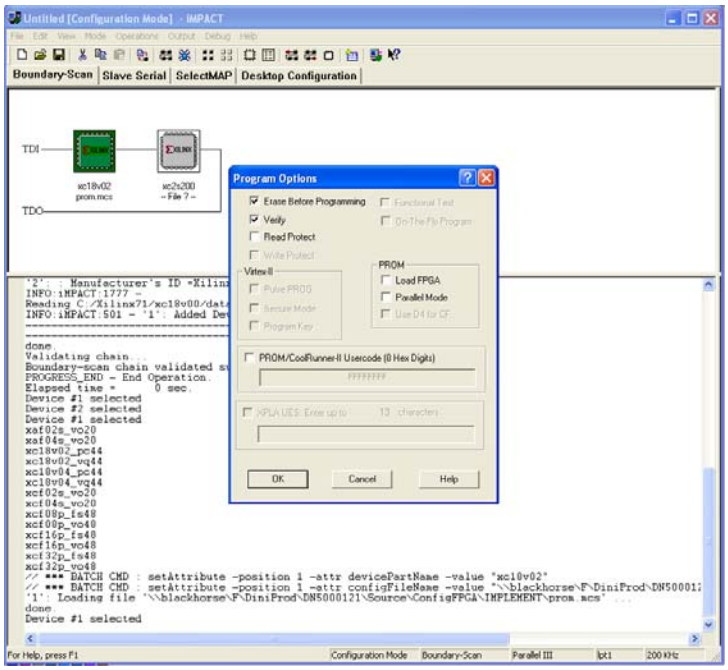



Figure 18 Impact Window

To program the prom. Right-click on the prom and select Program… from the popup menu. In the options dialog that follows, the options "Erase before programming" should be selected, and "Verify" should be deselected. Press OK. The programming process takes about 35 seconds over the parallel port.

Power cycle the DN8000K10PCI. The new firmware is now loaded. You can close impact and disconnect the Parallel IV cable.

Chapter

5

# Hardware

## 1 Overview

The DN8000K10PCI was designed to maximize the number of useful gates in your emulation project running at speed by providing the densest interconnect possible. To achieve this goal, the DN8000K10PCI is equipped with the highest-capacity FPGAs available today, the Xilinx Virtex 4 LX200. The FPGAs on the DN8000K10PCI are in the largest, 1513-ball package to give the user extremely high IO count, for high bandwidth and low-latency interconnect between FPGAs. Three hundred eighty nine differential links between FPGAs A and B allow for as much as 189 Gb/s communication between the two FPGAs.

In order to support enough bandwidth to deliver real time data to your design at speed, the DN8000K10PCI is equipped with an optional Xilinx Virtex 4 FX100 with RocketIO Multi-Gigabit Transceivers. Serial connections over Fibre, Coax ribbon cable, and Coax SMA cables allow for a total aggregate 150 Gb/s off-board communication.

Monitoring your design and supplying test vectors is simple with an onboard PCI interface bridge chip, capable of full 66Mhz, 64-bit PCI while reserving 100% of the FPGA fabric for user logic. The PCI interface is taken care of. You will be able to communicate to the DN8000K10PCI into a PCI slot right out of the box.

To allow you to connect the FPGA to the resources that will be on your end product, the DN8000K10PCI also has high speed expansion capabilities.

Below is a block diagram of the DN8000K10PCI

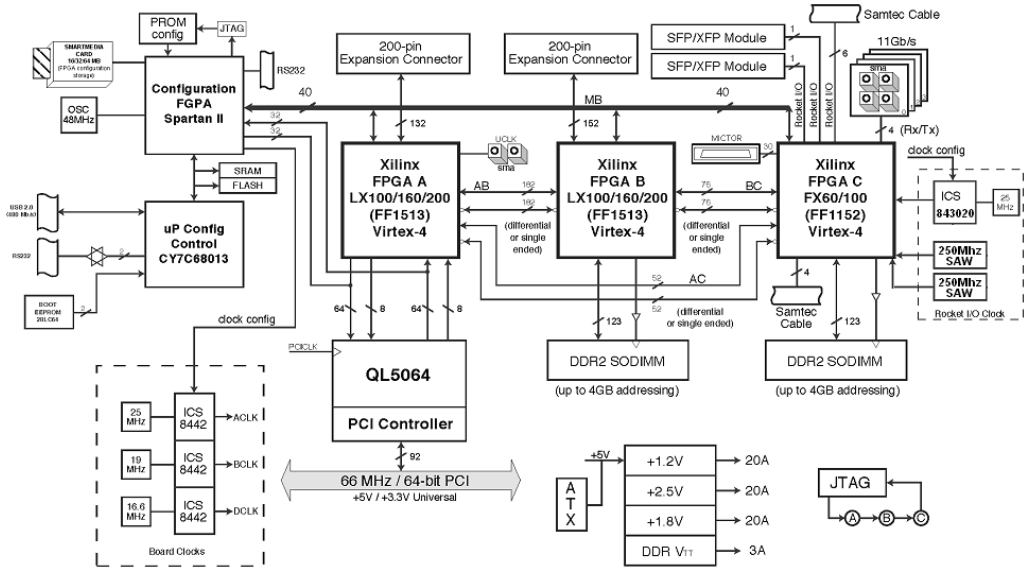


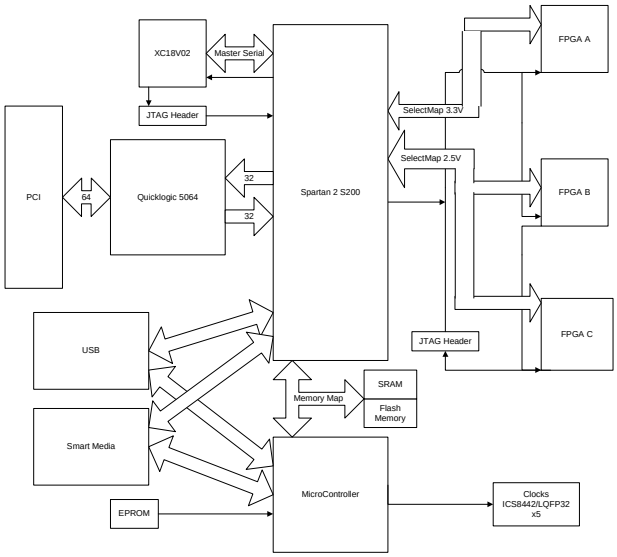

Figure 19 DN8000K10PCI Block Diagram

The following sections describe in detail each circuit on the DN8000K10PCI. Note that Schematics appearing in this section are illustrative and may have had details omitted or have been modified for clarity and brevity. If you need to probe, modify or design around the DN8000K10PCI you will need to examine the complete schematics. See the schematic pdf on the user CD. An assembly drawing has also been provided to help you find probe points on the DN8000K10PCI.

## 1.1 Overview

The goal of the configuration circuit on the DN8000K10PCI is to allow the user to configure his FPGAs using any host interface. The configuration system on the DN8000K10PCI allows configuration over PCI, USB, JTAG, or automatic configuration from a SmartMedia card.

# 2 Configuration Circuit




The circuit is designed to provide an easy configuration solution that will work out-of-the-box for most users. For special configuration requirements, the configuration circuitry is programmable. The Verilog code for the configuration FPGA and the C code for the microcontroller are both provided on the reference CD. The C code for the PCI controller program and USB Windows GUI controller program are also included on the User CD.

## 2.1 The Spartan 2 FPGA

The configuration circuitry of the DN8000K10PCI is built around a Xilinx Spartan II Fpga. The SelectMap interface of the user FPGAs is connected directly to the general purpose IOs of the Spartan 2, allowing the maximum flexibility of configuration. The Spartan 2 also shares connectivity with the three user FPGAs over a 40-bit Main bus, allowing fast transfers from a computer to the user design over PCI or USB. The Spartan 2 FPGA also provides IO expansion for the Cypress Microcontroller. The Spartan II FPGA comes preloaded with a core that provides a way to program the Virtex 4 FPGAs over PCI, USB and SmartMedia.

The Spartan FPGA is connected to the Cypress microcontroller's address and data busses, and the control registers within the Spartan II FPGA that control FPGA configuration are memory-mapped into the MCU's address space.

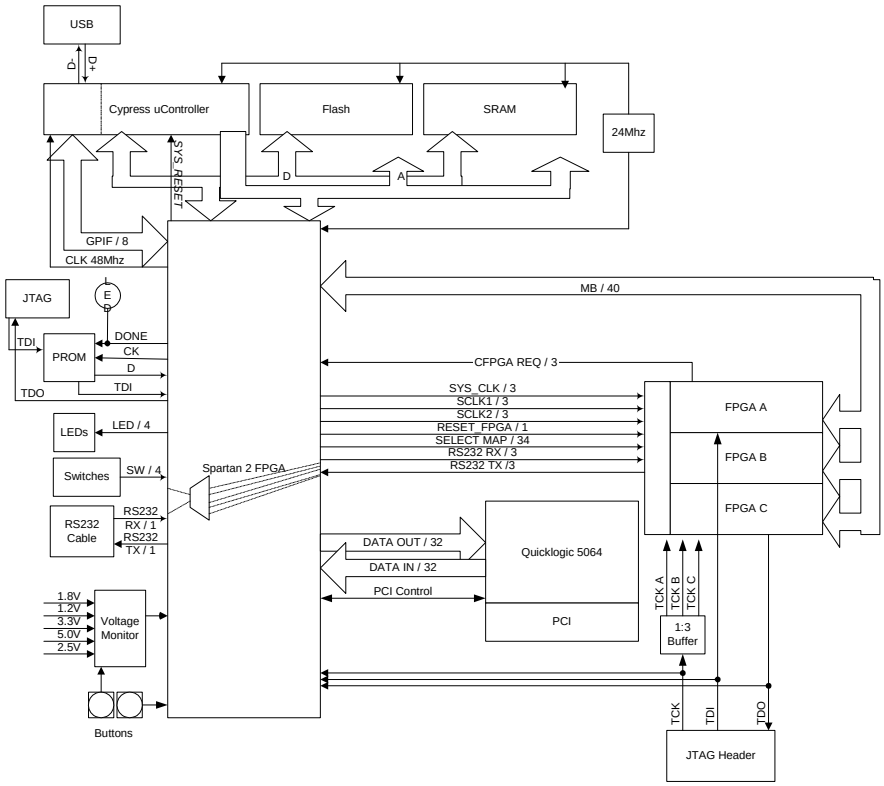



Figure 20 Spartan II IO Connections

### 2.1.1 Spartan Configuration

The Spartan 2 FPGA is configured from a Xilinx serial prom. The Spartan's configuration mode is hard-wired into Master Serial mode. After power up, the Spartan automatically clocks an external PROM, U41, which programs the FPGA over the serial configuration data pin DIN.

A green LED, DS24, lights when the DONE pin is high. This signal is driven by the Spartan 2 FPGA when it is configured and running.

Both the Spartan and the serial prom are connected in a JTAG chain attached to J14. This header is used when performing firmware updates to update the PROM.

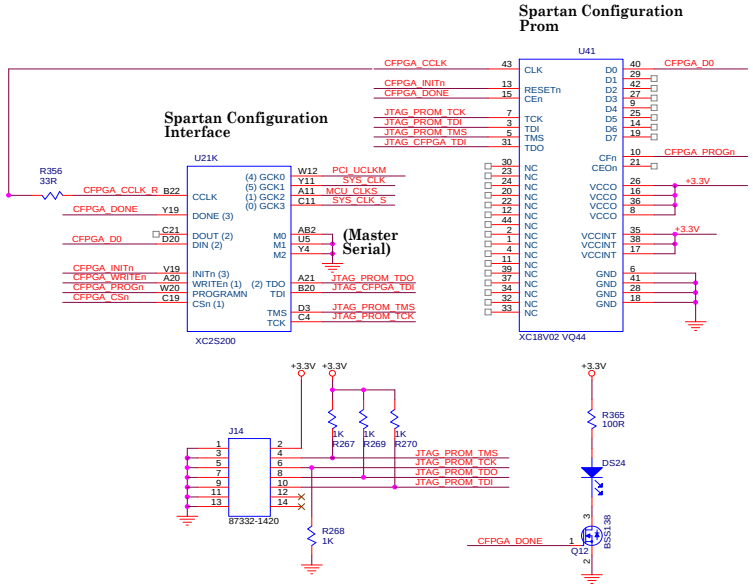



Figure 21 Spartan II Configuration

As soon as the Spartan II FPGA is configured, it resets the Cypress microcontroller. Pull-downs on the PROG pin of FPGAs A B and C ensure that the FPGAs cannot be active unless the Spartan II is successfully configured.

### 2.1.2 Smart Media

The Smart Media card interface is connected to the IOs of the Spartan 2 FPGA.

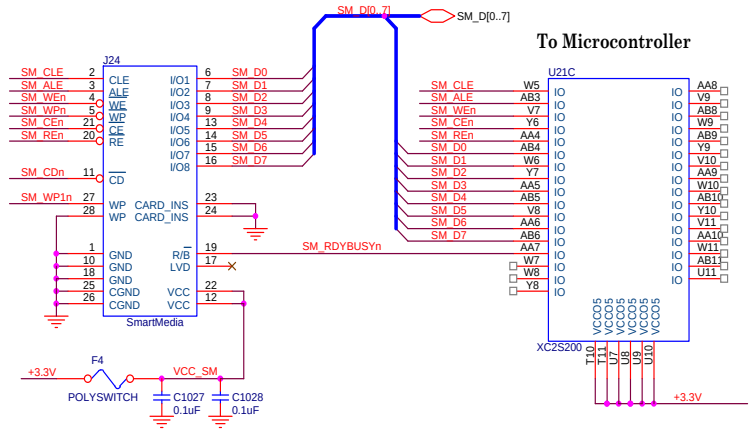



Figure 22 Smart Media interface

The Smart Media data bus, D[0-7], also connects to the microcontroller. Currently the MCU connection is not used. The Microcontroller is able to read from the Smart Media interface by accessing the Spartan's memory-mapped data over the MCU memory interface for the purposes of reading instructions from SmartMedia cards.

For instructions on creating a Smart Media card for configuring the DN8000K10PCI, see the section *Configuration Options: Smart Media.*

### 2.1.3 MCU communication

The MCU communicates to the Spartan 2 FPGA over it's external memory interface, pins D0:7 and A0:15. The Spartan 2 is assigned the address range 0xDF00 to 0xDFFF in the Microcontroller's memory space.

The 480Mbs data rate of USB 2.0 is too fast for the microcontroller to control, so the MCU's hardware passes USB bulk transfer data to the MCU GPIF interface. These signals, SM[0-7] and GPIF_CTL, GPIF_RDY, connect to the Spartan FPGA. The SM[0-7] signals also connect to the SmartMedia card socket, although the MCU does not communicate with the SmartMedia interface directly. The MCU_IFCLK signal provides a clock for this interface. The signal is driven from the Spartan 2 FPGA.

### 2.1.4 PCI communication

To enable configuration over PCI, the Spartan 2 is connected to a subset of the Quicklogic 5064 PCI interface. Due to the available number of IOs on the Spartan 2 FPGA, only 32 of the 64 data bits in the Quicklogic interface are connected to IOs on the Spartan 2. Also, the Spartan 2 does not connect to any of the signal required for DMA operation over PCI.

From the PCI host perspective, the address range BAR0 is directed to the Spartan 2 FPGA, and BAR1-BAR7 is directed towards the Virtex 4 FPGA A.

Since FPGA A and the Spartan 2 both can access the Quicklogic 5064 back-end, only one must communicate with the QL5064 at a time. The signals SP_PCI_REQ and LX_PCI_ACK are used to control communication with the QL5064.

### 2.1.5 RS232

The DN8000K10PCI has two RS232 headers. One (P2) is used by the microcontroller unit to provide configuration feedback and control. The other (P1) is connected to the Spartan 2 FPGA. The Spartan 2 FPGA has one RX and one TX signal connected to each Virtex 4 FPGA. The Spartan FPGA will multiplex the RX and TX signals to the Virtex FPGAs to the RS232 header P1. The Spartan 2 internally multiplexes the signals on the user RS232 header P1, to one of these three sets of signals. To change the Virtex 4 FPGA that has access to the RS232 headers, you can use the provided USB application program, or you can change the setting on a terminal connected to the Microcontroller unit's RS232 port (P2).

Since RS232 uses a 12V signal levels, the RS232 signals from the SpartanII are first buffered through a voltage translation buffer shown below.

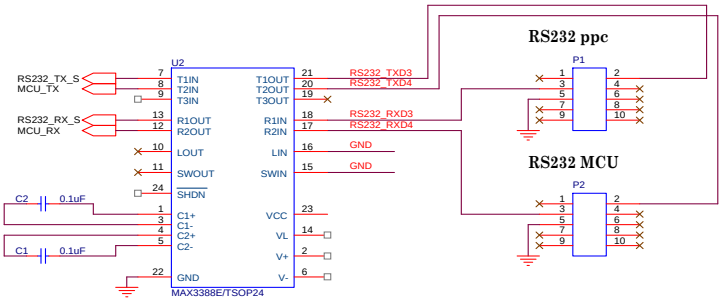



Figure 23 RS232 buffer

On the back side of the DN8000K10PCI, there are two duplicate RS232 ports (P7 and P8) that can be used if an installed daughter card is covering the headers on the front. These duplicate headers are not installed by default, but can be installed on request. They are compatible with a surface mount, 5x2 0.1" header.

### 2.1.6 IIC

There is a single IIC bus on the DN8000K10PCI connecting all IIC enabled chips on the board. On this bus are three MAX1617A temperature sensing chips (U3, U4, U24), two DDR2 SODIMM sockets, and a serial EPROM. The temperature sensors on the IIC bus are polled about once per second by the MCU to read the temperature of each FPGA.

## 2.2 Configuration Options

The DN8000K10PCI allows FPGA configuration from any of four methods.

When a Virtex 4 FPGA is configured, the DONE pin on the FPGA is pulled high. The DN8000K10PCI has a green LED attached to the DONE signal of each to indicate the state of the DONE pin on the three Virtex 4 FPGAs and on the SpartanII configuration FPGA.

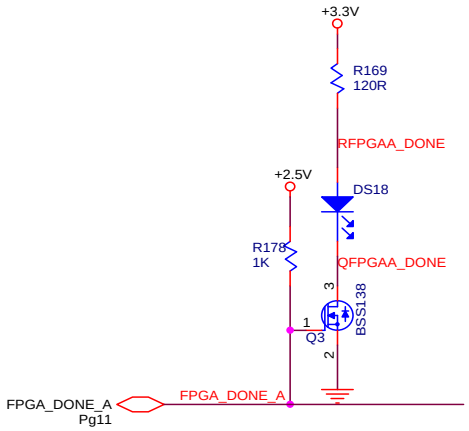



Figure 24 DONE LEDs

### 2.2.1 Jtag

Jtag is the only configuration method on the DN8000K10PCI that does not use the Virtex 4 SelectMap configuration interface. When programming the user FPGAs over a JTAG cable plugged into J13, the DN8000K10PCI configuration circuitry is not used.

A JTAG connection is required to use some Xilinx configuration tools like ChipScope, and readback from Impact. Configuration over JTAG is slower than SelectMap. You can still use the SmartMedia or USB interfaces to control clock settings if you plan to configure through JTAG.

To configure using JTAG, we recommend using Xilinx Parallel cable IV, or Xilinx platform USB cable. The Xilinx program. You should set the configuration speed of your JTAG cable to 4Mhz or below.

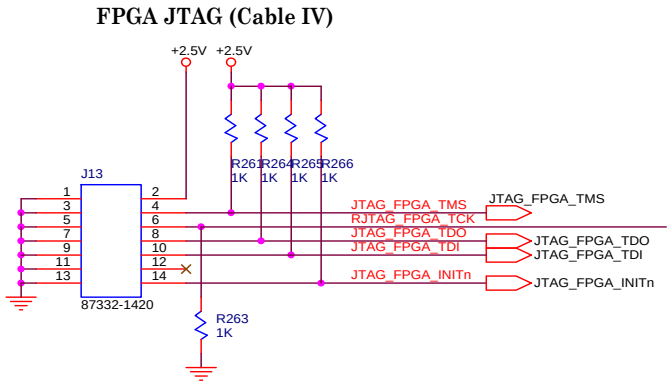



Figure 25 FPGA JTAG Header

The JTAG signals TMS is bussed to all three Virtex 4 FPGAs. TDO connects to FPGA A, the TDO of FPGA is connected to TDI of FPGA B, the TDO of FPGA B connects to the TDI of FPGA C and TDO of FPGA C is connected to the TDI of J13. TCK is buffered and passed to each FPGA in a point-to-point fashion.

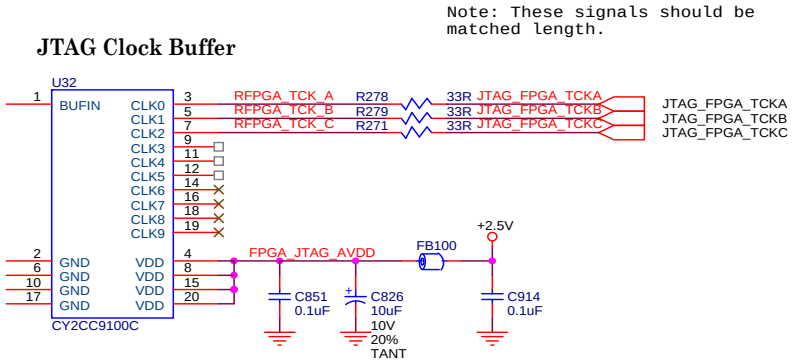



Figure 26 TCK buffer

The INITn signal is not used.

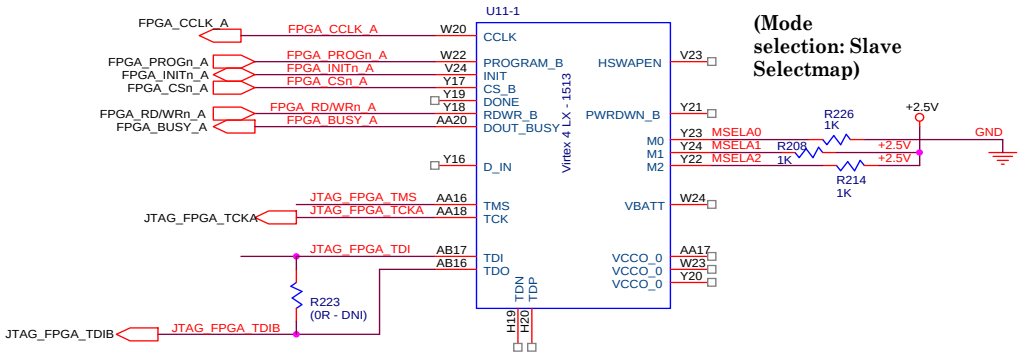



Figure 27 FPGA A Configuration Bank

If you ordered your DN8000K10PCI with one or more FPGAs not installed (Option FPGA A NONE, FPGA B NONE, or FPGA C NONE) then a bypass resistor is installed connecting the TDI pin to the TDO pin of the uninstalled FPGA. This is so the JTAG chain will remain intact when FPGAs are missing.

### 2.2.2 SmartMedia

When the DN8000K10PCI powers on, the microcontroller reads the contents of any SmartMedia card that is in the SmartMedia slot. The microcontroller by default opens a file on the root directory named "Main.txt" if it exists. This file contains instructions for the configuration circuitry to configure the Virtex 4 FPGAs.

To create a SmartMedia card to control the DN8000K10PCI configuration, insert the SmartMedia card into a card reader (provided) and connect it to a PC. Create a file on the root directory of the card and call it "Main.txt"

In main.txt, write a series of configuration commands, separated each by a new line. A valid command is one of the following:

```
// <comment>
FPGA A:<filename>
FPGA B:<filename>
FPGA C:<filename>
CLOCK FREQUENCY: <clockname> N <number> M <number>
SANITY CHECK: <yn>
VERBOSE LEVEL: <level>
RS232: <portnumer> <fpganame>
MEMORY MAPPED 0x<SHORTADDR> 0x<BYTE>
MAIN BUS 0x<WORDADDR> 0x<WORDDATA>
```

<comment> can be any string of characters except for newline.
<filename> can be the name of a file on the root directory of the SmartMedia Card.
<number> can be any one or two digit positive integer in decimal
<clockname> can be [A,B,D,2] A is ACLK, B is BCLK, D is DCLK and 2 is the RocketIO clock synthesizer.
<yn> can be the letter y or the letter n
<level> can be 0,1,2 or 3
<portnumber> can be 1,2,3, or 4. The Dn8000K10PCI only has 1 user RS232 port (1) so 2-4 will cause no operation.
<fpganame> can be [A,B,C,D,E,F,G,H,I]. The DN8000K10PCI only has 3 FPGAs (A,B,C), so D-I will cause the RS232 port to not function.
<SHORTADDR> is a 2-digit hex number (16 bits)
<BYTE> is a 1-digit hex number (8 bits)
<WORDADDR> 4-digit (32 bit) hex number representing a main bus address
<WORDDATA> 4-digit (32 bit) hex number containing data for a main bus transaction

The following table describes the function of each of the available main.txt commands.

| **Instruction** | **Function** |
|---|---|
| // <comment> | The MCU performs no operation and moves to the next command. |
| VERBOSE LEVEL: <level> | This command will set the amount of output the MCU will produce over the RS232 port during configuration. When level is set to 0, the MCU will produce only error output. Before this command is executed, the level is set to the default value 3. |
| FPGA A:<filename> | The Virtex 4 FPGA "A" will be configured with the file named by <filename> |
| FPGA B:<filename> | The Virtex 4 FPGA "B" will be configured with the file named by <filename> |
| FPGA C:<filename> | The Virtex 4 FPGA "C" will be configured with the file named by <filename> |
| SANITY CHECK: <yn> | If <yn> is set to y, then the MCU will examine the headers in the .bit files on the SmartMedia card before using them to configure each FPGA. If the target FPGA annotated in the .bit file header is not the same type as the FPGA the MCU detects on the board, it will reject the file and flash the error LED.<br><br>Before this command is executed, <yn> is set to the default value y.<br><br>If you want to encrypt of compress your bit files, you will need to set <yn> to n. Encrypting bit files is not supported or recommended by Dini Group. Previous revisions of Xilinx parts have been vulnerable to permanent damage caused by bugs in the encryption circuitry. |
| MAIN BUS 0x<WORDADDR> | Writes data in <WORDDATA> to the address on the main bus interface at |

| | |
|---|---|
| 0x<WORDDATA> | <WORDADDR>. This command only makes sense in the context of the Dini Group reference design, unless your design implements a compatible controller on the main bus pins. The Reference Design chapter contains a distription of this interface. See the board netlist or the reference design UCF for pin placement. |
| CONFIG REG 0x<SHORTADDR> 0x<BYTE> | Writes to an address in the MCU XDATA memory space. |
| RS232: <port> <fpga> | The RS232 port (P1) will be controlled by the FPGA <fpga> if <port> is 1 |
| CLOCK FREQUENCY: <clockname> N <number> M <number> | The MCU will adjust the clock synthesizer producing clock <clockname> to multiply it's reference frequency by <M> and divide it by <N><br><br>Note that the clock synthesizers have a limited bandwidth, and for clocks A B and D, the reference frequency * M must fall in the range 250Mhz-700Mhz. For clock 2 (RocketIO), reference * M must fall between 540 and 680Mhz. See datasheets for parts ICS8442AY and ICS843020-01<br><br>The reference frequencies are<br>ACLK 25Mhz<br>BCLK 14.18Mhz<br>DCLK 16Mhz<br>2CLK 25Mhz |

An example main.txt file:

```
VERBOSE LEVEL:0
// This will prevent the MCU output over RS232 to speed up configuration
FPGA A:a.bit
//this will load the configuration a.bit into FPGA A
CLOCK FREQUENCY: A N 4 M 10
// This will cause Aclk frequency to be
// 25*10=250 / 4 = 62.5Mhz
MAIN BUS: 0x0000 0x0001
//Writes to a register in FPGA A.
```

Even if you are not planning to configure your Virtex 4 FPGAs using a SmartMedia card, you may want to leave a SmartMedia card in the socket to automatically program your global and rocketIO clock. (Clocks may also be programmed using the provided USB application, or over the MCU RS232 terminal.)

### 2.2.3 USB

The USB interface on the DN8000K10PCI is provided by the Cypress microcontroller unit. The Cypress microcontroller is programmed to interrupt when it receives a USB vendor request.

When the MCU receives over USB a Bulk Transfer type request, it does not interrupt. The raw data contained in the bulk transfer is driven out on the GPIF pins of the MCU (the SM[0-7]

signals) to the Spartan 2. The data is clocked out using the MCU_IFCLK clock signal to the Spartan 2. As long as the signal GPIF_CTL is held high by the MCU, the Spartan 2 clocks MCU_IFCLK to receive the USB data.

When data is written to the Spartan 2 from a bulk transfer over the MCU's GPIF interface, the Spartan 2 either writes that data onto the SelectMap interface of the Vitex4 FPGAs, or onto the Main bus using the Main Bus interface described in the *Reference Design* chapter.

The control register FPGA_SELECT within the Spartan 2 determine to which interface this data is routed to.

### 2.2.4 PCI

The PCI interface on the DN8000K10PCI is primarily used directly by the Virtex 4 FPGA. The Quicklogic 5064 implements the PCI interface and delivers the data directly over the QL interface to FPGA A. However, when the host machine makes a read or write transaction to the DN8000K10PCI on a BAR0 address over PCI, the data is instead delivered to the Spartan 2 configuration FPGA. This allows the Spartan 2 to configure FPGAs, change other configuration settings, or communicate to FPGAs B and C over PCI.

The program AETest, supplied on the User CD along with the AETest source code, allows the user to configure FPGAs and change configuration settings over PCI. When used with the Dini Group reference design, or if the user has created a compatible Main Bus interface, the AETest program can also communicate directly to the FPGA A, B and C designs.

#### 2.2.4.1 Configuration

To configure an FPGA over PCI, the host program writes the following instructions to BAR0 of the DN8000K10PCI.

First, the host instructs the Spartan 2 FPGA to "select" an FPGA for configuration. Write one word of data to BAR0, 0x0208. The data 0x11 represents FPGA A, 0x12 is FPGA B, 0x13 is FPGA C.

Next, the host instructs the Spartan 2 FPGA to assert the PROGn signal (Reset) of the selected FPGA's SelectMap interface. This causes the FPGA to un-configure regardless of the reconfiguration setting made by the bit file's reconfigure setting. The PROGn signal must be asserted once before the FPGA is configured for the first time. Write a word to the address BAR0, 0x208. The data word 0x11 represents FPGA A, 0x12 is FPGA B, 0x13 is FPGA C.

When the FPGA is read to configure, it pulls the SelectMap signal INTn low. The configuration process should pause until this occurs. To read the current value of INTn on the selected FPGA, read from the BAR0 address 0x208. Bit 5 represents the value of the selected INTn signal.

After the INTn signal is detected, the Host should de-assert PROGn (Reset). Write to BAR0 address 0x208 the data word representing the selected FPGA. 0x11 is FPGA A, 0x12 is FPGA B, 0x13 is FPGA C.

The configuration stream for the FPGA is then sent to BAR0, address 0x210, one byte at a time. Some time during the configuration stream byte loading process, a startup sequence is sent to the FPGA and the FPGA becomes operational. This startup sequence is contained within the bit file.

To determine of the selected FPGA is currently configured (i.e. configuration was successful), read from BAR0 address 0x208. The bit 5 contains the state of the DONE FPGA pin, the bit 6 contains the state of the FPGA INIT signal.

By convention, the host program should leave the Spartan in the FPGA deselected state. To deselect the FPGA, write to BAR0, address 0x208 the data 0x10. (FPGA SELECT NONE)

#### 2.2.4.2 Config Space

PCI can also be used to control other configuration functions on the DN8000K10, such as temperature sensors and clocks. This is done by altering the data in the XDATA memory space of the configuration MCU.

To write to the MCU address space, access the DN8000K10's BAR0 at the address MCU_BAR_ADDR 0x258. Send a 32-bit word of data. This data is decoded as follows

Bits 31-16: address in the XDATA space. (only addresses 0xDF00-0xDFFF reside in the Spartan 2)

Bits 15-8: Ignored

Bits 7-0: The Data to write

To read from the MCU address space, access the DN8000K10's BAR0 at the following 32-bit address:

Bits 31-24: The DN8000K10PCI's BAR0

Bits 23-16: the lower 8 bits of XDATA address you would like to read. This corresponds to addresses 0xDF00-0xDFFF of the XDATA address. (Only addresses 0xDF00-0xDFFF reside in the Spartan 2)

Bits 15-0: 0x0260

#### 2.2.4.3 Main Bus Space

PCI can also be used to send information to and from your Virtex 4 user design through the Spartan 2 FPGA. Communication directly to the user design can also be accomplished from

FPGA A by communicating directly with the QL PCI backend interface. This method is an order of magnitude faster, and allows the use of advanced PCI features like DMA. See *Hardware: PCI interface*. Communication with the FPGAs through the Spartan occurs using the Main Bus interface. For information on the main bus interface see *Reference Design: Main Bus Interface.*

To write to the main bus interface, write to BAR0 address QLPCI_REG_MBADDR with the 32-bit value representing the main bus address you would like to write to. Then, write a second PCI write to address QLPCI_REG_MBWRDATA with 32-bit data representing the data that you would like to write to main bus. After the Spartan 2 has received a write to both the MBADDR and MBWRDATA registers, it will write to the main bus interface.

To read from the main bus interface, first write to BAR0 address QLPCI_REG_MBADDR with the 32-bit value representing the main bus address you would like to read from. Then, read from BAR0, QLPCI_REG_MBRDDATA. The returned value will be the value read off the main bus at the selected address. When an error has occurred (No FPGA responded to the read request) the Spartan will return the value 0xABCDABCD.

## 2.3 FPGA configuration Process

For information regarding the JTAG interface and configuration, See Xilinx publication UG071, Virtex 4 configuration guide.

When configuring over PCI, USB or SmartMedia, the FPGAs are configured over the Virtex 4 SelectMap bus.

All SelectMap signals are connected directly to the Spartan2 FPGA. The SelectMap signals are:

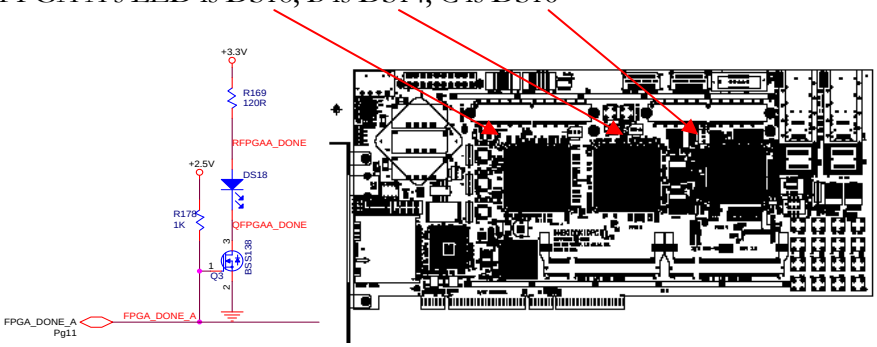
| | |
|---|---|
| D[0-7] | SelectMap data signals. |
| PROGRAM_B | Active low asynchronous reset to the configuration logic. This will cause the FPGA to become unconfigured. The documentation refers to this signal as PROGn |
| DONE | After the FPGA is configured, it is driven high by the FPGA. |
| INIT | Low indicates that the FPGA configuration memory is cleared. After configuration, this could indicate and error. |
| RDWR_B | Active low write enable. The Documentation refers to this signal as RDWR |
| BUSY | When busy is high, the SelectMap configuration stream must stop until BUSY goes low. |
| CS_B | SelectMap chip select. The documentation refers to this signal as CSn |
| CCLK | Signals D[0:7], DONE, RDWR_B and CS_B are clocked on CCLK |

Each Virtex 4 FPGA has a complete set of SelectMap signals connected point-to-point to the Spartan 2, except for FPGA B and C, who share signals D[0-7]. All signals are 2.5V CMOS signals except for D[0-7] of FPGA A (Signals SELECTMAP_3V_D[0-7]), which are 3.3V CMOS.

All commands required to configure a Virtex 4 FPGA are created and embedded in the .bit files created by the Xilinx Bitgen program. The DN8000K10PCI does not interact with the SelectMap interface other than to reset the FPGA using the PROGn-INTn-PROGn reset sequence described in UG071, and to copy a bit stream file unaltered to the FPGA over the data pins D[7-0]. Select map commands can be issued to the Virtex 4 FPGA from the host using the same interface used to configure and FPGA.

After a Virtex 4 FPGA is configured, it asserts the signal DONE. On the DN8000K10PCI, these signals have an LED attached to each DONE signal placed near the upper corner of each FPGA.

FPGA A's LED is DS18, B is DS14, C is DS16




If your Virtex 4 FPGA design is failing to produce the intended (or any) results, you should check the DONE light above the FPGA to make sure it is configured correctly. The design files created by Xilinx bitgen software contain a CRC check, so if the Virtex 4 FPGA detects a CRC failure, there was a transmission error during configuration and the DONE light will not glow. The DN8000K10PCI microcontroller also checks the design files you send to make sure they are compiled for the FPGAs that are installed on your board. If they are not, then the microcontroller unit halts the configuration process. As a result, when the DONE light goes on, you will know that the configuration process was successful.

## 2.4 MCU

The operation of the Spartan II is monitored and controlled by a Cypress CY7C68013 microcontroller. The microcontroller also has a USB 2.0 interface that can be used to monitor the board, control configuration, or transfer data to and from the user FPGA design. Basic operation can be controlled over an RS232 link from a computer terminal.

### 2.4.1 RS232

The primary method of user interaction with the DN8000K10PCI configuration circuitry is the MCU's RS232 port (P2). The Cypress CY7C68013 has two RS232 pins that are buffered through a 12V voltage translation buffer for use with a standard computer serial port.

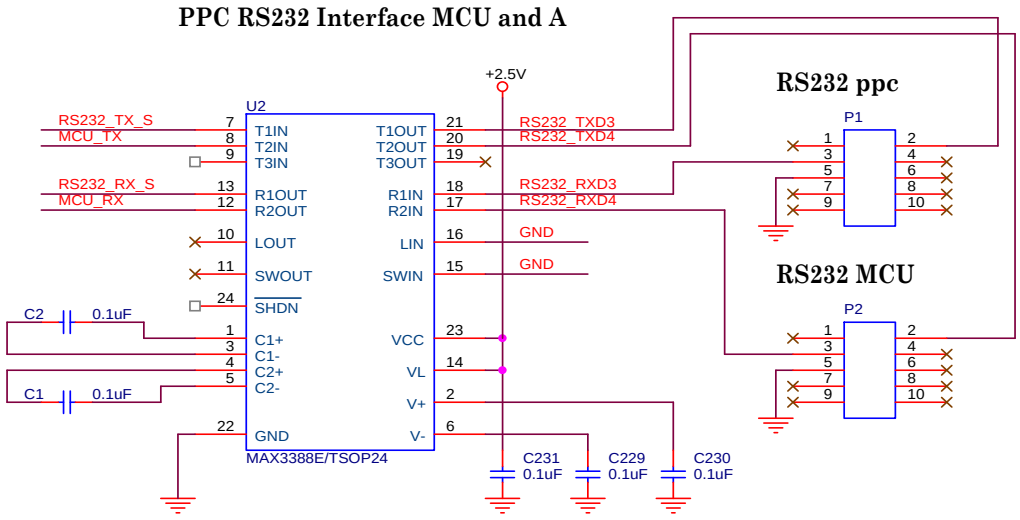



Figure 28 RS232 Buffer and Headers

The RS232 port will be able to communicate with a standard PC serial port set to 19200 baud, 8 data bits, no parity, no handshaking. When you connect a computer terminal to the port and power on the DN8000K10PCI, the firmware loaded on the microcontroller unit will display a menu on the terminal. This menu will allow you to control the basic configuration options of the DN8000K10PCI including configuration, clock frequencies, and the Virtex 4 FPGA RS232 ports.

### 2.4.2 Clocks

The Cypress CY7C68013 is also responsible for configuring the global clocks and RocketIO clock of the DN8000K10PCI. The Cypress CY7C68013 MCU reads the file "main.txt" from the SmartMedia card in the socket (J24), and follows the users clock configuration commands.

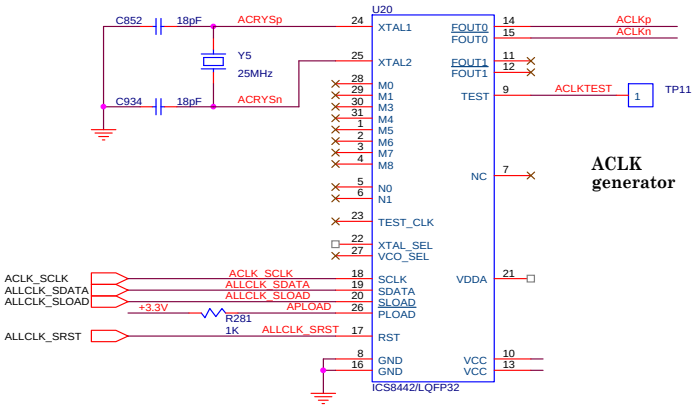



Figure 29 8442 Clock synthesizer

The 3 ICS8442 clock synthesizers on the DN8000K10PCI used for generating the global clocks, ACLK, BCLK and DCLK, share a serial configuration bus connected to the MCU to program them. The ICS8442 frequency synthesizers are capable of multiplying and dividing the reference frequencies provided by their reference crystals. The MCU loads the user's desired multiplication "M" value, and division, "N" value into the settings registers in the ICS8442 chip.

### 2.4.3 LEDs

The MCU is connected to 4 red LEDs that are visible from outside the PC case when the DN8000K10PCI is plugged into a PCI slot. The LEDs flash a status code during and after configuration.

All four flashing LEDs means there has been an error configuring at least one FPGA.

### 2.4.4 Memory space

The XDATA memory space of the MCU is partitioned into four sections.

| | |
|---|---|
| 0x0000 - 0x1FFF | internal data/program memory |
| 0x2000 - 0xCFFF | external SRAM |
| 0xDFF0 - 0xDFFF | memory mapped registers (no external memory accesses) |
| 0xE000 - 0xFFFF | reserved by MCU, RD/WR strobes not active in this region |

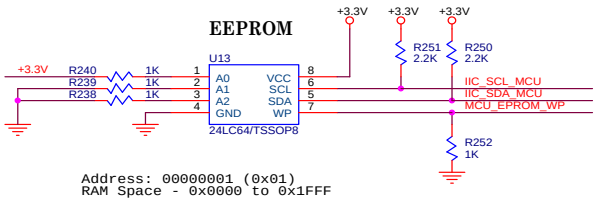
The internal data memory region is mapped to an internal SRAM in the Cypress MCU. When the microcontroller code calls memory access from this region, the external Address and Data busses are not used. After power on reset, the MCU reads from the IIC EPROM connected to the MCU_EPROM signals and fills this internal memory before allowing the PC to run. The code in this section of memory contains core functions of the Dini Group firmware, like setting up the interrupt registers, communicating with USB, and allowing firmware updates.

The external SRAM is used for heap data.

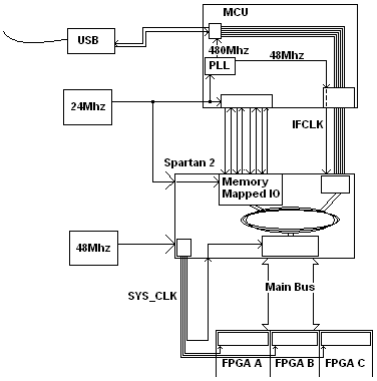
The memory mapped register region (The DF region) contains registers in the Spartan 2 FPGA that control FPGA configuration.

The program memory space of the MCU is directly mapped to the external Flash memory.

When the Cypress MCU is reset (which happens after the Spartan 2 is configured), it loads its boot code into its 8kB of internal memory from a serial EPROM (U13). The code in the EPROM instructs the MCU to execute code located on the FLASH memory (U19). The code in the EEPROM and FLASH is located on the user CD.

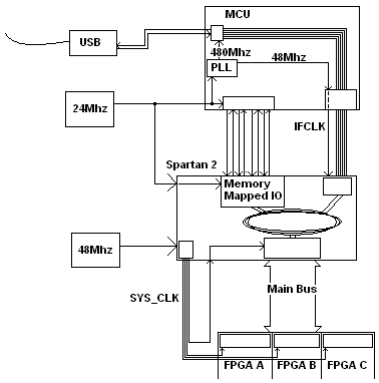



Communication over the MCU memory bus to the Spartan 2 is synchronized to the 24Mhz MCU_CLK (X3). For information regarding the timing of transactions on this bus, see the Cypress CY7C68013 user manual.




The Configuration FPGA is connected to the MCU_DATA[7:0] signals, the MCU_ADDR[15:0] signals and the MEM_OE signal, allowing it to decode address accesses of the MCU. The Configuration FPGA is programmed to respond to accesses in the XDATA address space in the address range of 0xDF00 to 0xDFFF

Communication over the MCU memory bus to the Config FPGA is synchronized to the 24Mhz MCU_CLK (X3). For information regarding the timing of transactions on this bus, see the Cypress CY7C68013 user manual.




The following registers implemented in the Configuration FPGA are accessible as part of the MCU's XDATA address space.

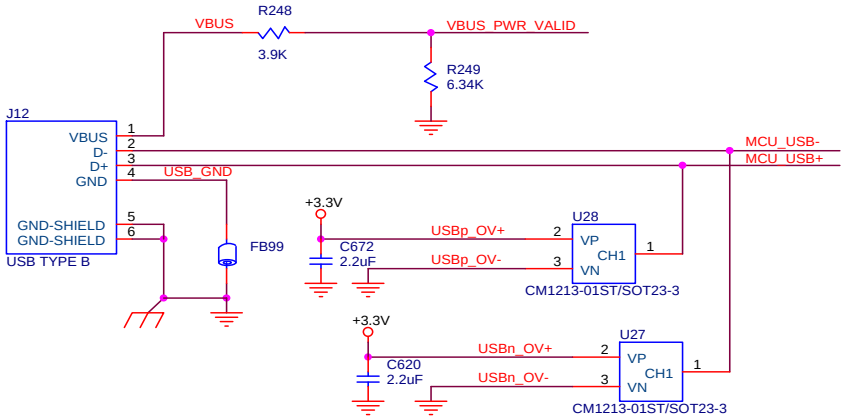
| **Register Name** | **XDATA Address** | **Description** |
|---|---|---|
| DATA | DF00 | Used when reading from SM but not configuring |
| COMMAND | DF01 | Commands for the SM |
| ROW_LADDR | DF02 | Holds lower 8-bits of SM address |
| ROW_HADDR | DF03 | Holds upper 8-bits of SM address |
| ROW_XADDR | DF04 | Holds extra bits of SM address |
| NUM_BYTES_0 | DF05 | Holds lower 8-bits of the number of bytes to read |
| NUM_BYTES_1 | DF06 | Holds upper bits of number of bytes to read in |
| BITS_1 | DF07 | BIT7: mcu_fpga_config_rd BIT6: |
| BITS_2 | DF08 | BIT4: FPGA_DONE BIT3 CPLD_idle BIT2: |
| SM_SIGNALS | DF09 | |
| MCU_XADDR | DF0A | Address register for upper FLASH/SRAM bits |
| MCU_CNTL | DF0B | Address register for upper FLASH/SRAM bits |
| FPGA_SELECT | DF0C | FPGA_select[5:0] = bits 5:0 |
| PPC_RS232_ABSELECT | DF0D | |
| PPC_RS232_CDSELECT | DF0E | |
| FPGA_CNTRL | DF0F | bits[1:0] = 01 (write address), 10 (data write), 11 |
| FPGA_BE | DF10 | select byte in addr, read, and data bytes |
| FPGA_RD_DATA | DF11 | |
| FPGA_WR_DATA | DF12 | |
| FPGA_ADDR | DF13 | |
| FPGA_ERROR | DF14 | |
| GPIF_DATA | DF20 | |
| GPIF_ERROR | DF21 | |
| HOLD_DONES | DF22 | |
| STATES | DF23 | [7:4] = GPIF_STATE, [3:0] = FPGA_STATE |
| FPGA_FREQ_H | DF24 | |
| FPGA_FREQ_SEL | DF25 | |

| | | |
|---|---|---|
| FPGA_FREQ_L | DF26 | |
| MCU_STUFFING1 | DF27 | |
| MCU_STUFFING2 | DF28 | |
| SERIAL_CLK_CTRL_0 | DF29 | |
| SERIAL_CLK_CTRL_1 | DF30 | |
| MB80_1_CTRL0 | DF36 | |
| MB80_1_CTRL1 | DF37 | |
| MB80_2_CTRL0 | DF38 | |
| FPGA_COMMUNICATION | DF39 | |
| MB80_2_CTRL1 | DF40 | |
| MB64_1_CTRL | DF41 | |
| MB64_2_CTRL | DF42 | |
| MB64_3_CTRL | DF43 | |
| CPLD_CS_N_CTRL | DF44 | |
| CPLD_DATA | DF45 | |
| CPLD_ADDR | DF46 | |
| GCLK_MSEL_CTRL | DF47 | |
| FPGA_PH0_DVAL | DF48 | |
| FPGA_PH1_DVAL | DF49 | |
| FPGA_PH2_DVAL | DF50 | |
| CF_REG_OFFSET | DFE | |
| NEW_CONFIG_VERSION | DFFD | |
| NEW_BOARD_VERSION | DFFE | |
| OLD_BOARD_VERSION | DFFF | |

These registers can be written to from the USB interface. See *USB Software: Programmers Guide*.

### 2.4.5 USB

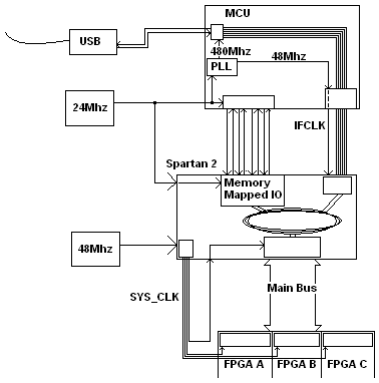
The Cypress CY7C68013 has a built-in USB 2.0 interface. The USB type B connector on the DN8000K10PCI (J12) is connected directly to the USB pins on the Cypress MCU.




**USB Transient Protection**

The USB protocol is completed by the Cypress CPU.

The Cypress receives a 24Mhz clock from an oscillator (X3). The Cypress internally multiplies this clock to 480Mhz for USB 2.0 and 48Mhz for GPIF operation. The core runs at 24Mhz along with the external memory interface. Communication over this external memory interface is clocked using the MCU_IFCLK signal driven from the MCU at 48Mhz. (The Spartan communicates over main bus with the Virtex 4 FPGAs using a separate 48Mhz oscillator (X1) and distributes this clock to each FPGA including itself)




### 2.4.6 Smart media

The SmartMedia card socket pins are bussed among the Cypress MCU GPIF pins, the Spartan 2 FPGA IOs, and the SmartMedia card socket. After reset, the MCU uses this connection to look for and read the contents of the file main.txt on the SmartMedia card. The main.txt file contains instructions for configuring the user design into the three Virtex 4 FPGAs.

After reading the configuration instructions, the MCU reads the headers of the user's FPGA design (".bit") files and verifies that they target the correct type of FPGA that are installed on

your DN8000K10PCI. This will prevent damage to the FPGA from an incorrect or corrupt .bit file. This behavior can be turned off.

If this check is passed, MCU uses its memory mapped interface with the SpartanII to instruct the SpartanII to read the SmartMedia card and configure the Virtex 4 FPGAs over SelectMap bus.

# 3 Clocking

The clocking circuitry on the DN8000K10PCI is designed for high-speed operation. The flexible clock design should meet the most difficult clocking needs, allowing 8 totally asynchronous, controllable clock sources for each FPGA.

All clocks operating above 100Mhz are fully differential, LVDS signaled, low skew, low jitter clocks.

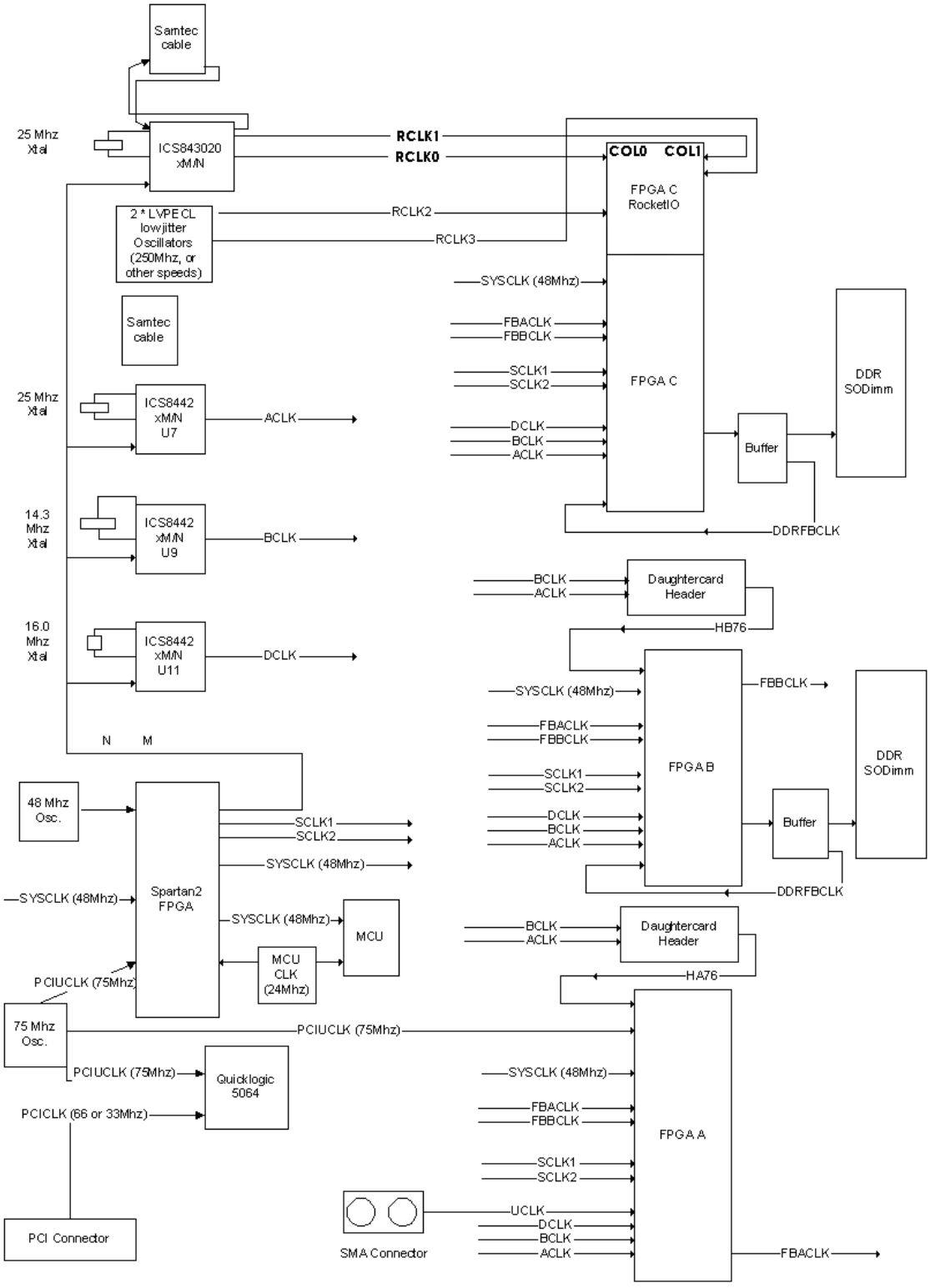



Figure 30 DN8000K10 clocking

From the above diagram, the global clocks are listed here.

RCLK1 – An ICS frequency synthesizer, either an ICS8442, ICS84321 (100-250Mhz), or ICS84020 (667Mhz). This clock is configured from the MCU using the USB controller or the SmartMedia card. This clock is supplied to MGT_CLK pins on FPGA C and can be used as an MGT reference clock for any MGT tile on the left column. The Synthesizer can also be configured to use an external clock input from the QSE-DP Samtec RocketIO connector J3.

RCLK2/3 – An Epson 250Mhz oscillator. This clock can be used to supply an MGT reference clock to FPGA C in either the right of left columns.

ACLK, BCLK, DCLK. These global clocks are supplied by ICS8442 frequency synthesizers. They are configured from the MCU to output a user-specified frequency from 31 to 700Mhz. They are each distributed to FPGAs A B and C. The clock signals are point-to-point from a LVDS clock buffer. The signal names in the schematic for these clocks are ACLKAp, ACLKAn, ACLKBp, ACLKBn, ACLKCp, ACLKCn, BCLKAp, BCLKBn, BCLKCp, BCLKCn, DCLKAp, DCLKAn, DCLKBp, DCLKBn, DCLKCp, DCLKCn

SCLK1/2 – These single-ended clocks run at low-speed and are controllable from the USB interface, allowing for software that controls single-stepping designs. Both clocks are delivered to FPGAs A B and C. The clock is sourced directly from the Spartan 2 configuration FPGA.

Sysclk – this 48Mhz, single-ended clock is driven from the configuration FPGA at a fixed frequency. It is delivered to FPGAs A, B, C and the configuration FPGA. This clock is used by the Dini Group reference design to clock the Main Bus interface.

MCU clk- this reference clock is used by the MCU to generate frequencies required for the USB protocol. It is not available to the user.

PCIUCLK – This single-ended, 75Mhz fixed clock is delivered to the Configuration FPGA, the Quicklogic PCI bridge, and FPGA A. It is used to clock the QL PCI back-end interface. It can also be used for the user design in FPGA A.

UCLK – This differential clock input is delivered to FPGA A.

FBACLK – This differential clock is driven from FPGA A and delivered to FPGA A, B and C. This clock can be used for controlled-clocks, odd clock division and multiplication, or forwarding a clock from on FPGA to another.

FBBCLK – This differential clock is driven from FPGA B and received at FPGA A, B and C.

HACLK – This differential clock is driven from the daughter card header A to FPGA A.

HBCLK – This differential clock is driven from the daughter card header B to FPGA B.

DDRACLK, DDRBCLK – This differential clock is driven by the FPGA to its associated DDR2 SODIMM header. A copy of the clock is externally buffered and the clock is received on the FPGA synchronized with its arrival at the SODIMM on the signal DDR_FBCLK.

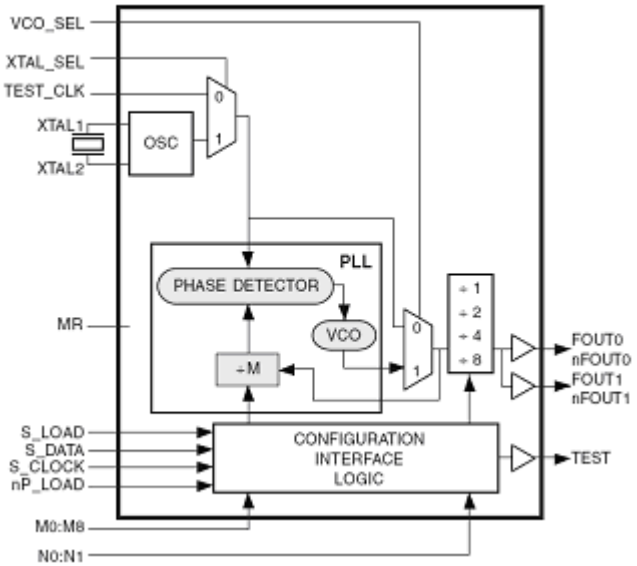
| Clock Name | FPGA A pin | FPGA B pin | FPGA C pin |
|---|---|---|---|
| ACLKp | N22 | AL20 | H19 |
| ACLKn | M22 | AL19 | H18 |
| BCLKp | J21 | AG20 | J14 |
| BCLKn | J20 | AF20 | K14 |
| DCLKp | L20 | AF19 | K19 |
| DCLKn | L19 | AF18 | J19 |
| FBACLKp | M21 | AH20 | H17 |
| FBACLKn | M20 | AH19 | J17 |
| FBBCLKp | L21 | AJ21 | J16 |
| FBBCLKn | K21 | AJ20 | J15 |
| UCLKp | P22 | - | - |
| UCLKn | P21 | - | - |
| SCLK1 | AG20 | L21 | AE18 |
| SCLK2 | AL20 | P22 | AF20 |
| HACLKA | AH18 | - | - |
| HACLKB | - | L20 | - |
| SYS_CLK | AF19 | M21 | AE21 |
| PCI_UCLKA | AH20 | - | - |

All of the signals listed above are point-to-point. That is the BCLKp pin on FPGA A and FPGA B are separate, equivalent signals.

## 3.1 Global Clocks

The three main global clocks are driven by ICS8442 clock synthesizers, each capable of producing frequencies from 30-500Mhz. The clock synthesizers can be programmed from a SmartMedia card, the Windows GUI application, or PCI. You may want to leave a pre-

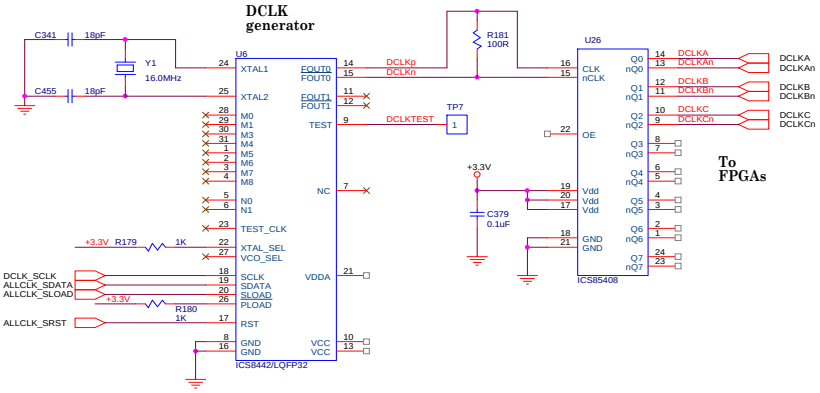
programmed SmartMedia card with a main.txt file on it to program your clocks, even if you intend to program the FPGAs over PCI or USB.




Each ICS8442 has an internal multiplication PLL that can operate between 250 and 700 Mhz. With 1, 2, 4, or 8x division on the output, the possible output frequencies are 31.25 – 700Mhz. (The maximum input frequency of the Virtex-4 FPGA is 500Mhz). VCO_SEL can be used to disable the PLL, so ACLK BCLK and DCLK can operate at their fundamental 25Mhz, 14.3Mhz and 16Mhz respectively.

The Serial configuration bus is connected to the Cypress MCU GPIF pins and controlled through software.

The crystal inputs are parallel resonant, fundamental mode.




The 8442 outputs are connected to a 1:8 LVDS buffer, and distributed to the FPGAs. Aclk and Bclk are also distributed to the expansion headers as well.

The clocks (ACLK, BCLK, DCLK) are LVDS, so a IGBUFDS module should be used to receive the clock differentially. External differential termination is not provided, so the DIFF_TERM attribute should be set on this module. See the example source code.

The crystal supplied for ACLK is 25.0 Mhz. With the 8442's PLL range, possible output frequencies are:

| | | | |
|---|---|---|---|
| 31.25 | 62.50 | 125.00 | 250.00 |
| 34.38 | 68.75 | 137.50 | 275.00 |
| 37.50 | 75.00 | 150.00 | 300.00 |
| 40.63 | 81.25 | 162.50 | 325.00 |
| 43.75 | 87.50 | 175.00 | 350.00 |
| 46.88 | 93.75 | 187.50 | 375.00 |
| 50.00 | 100.00 | 200.00 | 400.00 |
| 53.13 | 106.25 | 212.50 | 425.00 |
| 56.25 | 112.50 | 225.00 | 450.00 |
| 59.38 | 118.75 | 237.50 | 475.00 |
| 62.50 | 125.00 | 250.00 | 500.00 |
| 65.63 | 131.25 | 262.50 | |
| 68.75 | 137.50 | 275.00 | |
| 71.88 | 143.75 | 287.50 | |
| 75.00 | 150.00 | 300.00 | |
| 78.13 | 156.25 | 312.50 | |
| 81.25 | 162.50 | 325.00 | |
| 84.38 | 168.75 | 337.50 | |
| 87.50 | 175.00 | 350.00 | |

The crystal supplied for BCLK is 14.32 Mhz. With the 8442's PLL range, possible output frequencies are:

| | | | |
|---|---|---|---|
| 32.22 | 64.43 | 128.86 | 257.72 |
| 34.01 | 68.01 | 136.02 | 272.04 |
| 35.80 | 71.59 | 143.18 | 286.36 |
| 37.58 | 75.17 | 150.34 | 300.68 |
| 39.37 | 78.75 | 157.50 | 315.00 |
| 41.16 | 82.33 | 164.66 | 329.31 |
| 42.95 | 85.91 | 171.82 | 343.63 |
| 44.74 | 89.49 | 178.98 | 357.95 |
| 46.53 | 93.07 | 186.13 | 372.27 |
| 48.32 | 96.65 | 193.29 | 386.59 |
| 50.11 | 100.23 | 200.45 | 400.90 |
| 51.90 | 103.81 | 207.61 | 415.22 |
| 53.69 | 107.39 | 214.77 | 429.54 |
| 55.48 | 110.96 | 221.93 | 443.86 |

| | | | |
|---|---|---|---|
| 57.27 | 114.54 | 229.09 | 458.18 |
| 59.06 | 118.12 | 236.25 | 472.49 |
| 60.85 | 121.70 | 243.41 | 486.81 |
| 62.64 | 125.28 | 250.57 | |
| 64.43 | 128.86 | 257.72 | |
| 66.22 | 132.44 | 264.88 | |
| 68.01 | 136.02 | 272.04 | |
| 69.80 | 139.60 | 279.20 | |
| 71.59 | 143.18 | 286.36 | |
| 73.38 | 146.76 | 293.52 | |
| 75.17 | 150.34 | 300.68 | |
| 76.96 | 153.92 | 307.84 | |
| 78.75 | 157.50 | 315.00 | |
| 80.54 | 161.08 | 322.16 | |
| 82.33 | 164.66 | 329.31 | |
| 84.12 | 168.24 | 336.47 | |
| 85.91 | 171.82 | 343.63 | |

The crystal installed to reference DCLK is 16.0Mhz. Within the 8442's PLL range, the possible output frequencies are:

| | | | |
|---|---|---|---|
| 32.00 | 64.00 | 128.00 | 256.00 |
| 34.00 | 68.00 | 136.00 | 272.00 |
| 36.00 | 72.00 | 144.00 | 288.00 |
| 38.00 | 76.00 | 152.00 | 304.00 |
| 40.00 | 80.00 | 160.00 | 320.00 |
| 42.00 | 84.00 | 168.00 | 336.00 |
| 44.00 | 88.00 | 176.00 | 352.00 |
| 46.00 | 92.00 | 184.00 | 368.00 |
| 48.00 | 96.00 | 192.00 | 384.00 |
| 50.00 | 100.00 | 200.00 | 400.00 |
| 52.00 | 104.00 | 208.00 | 416.00 |
| 54.00 | 108.00 | 216.00 | 432.00 |
| 56.00 | 112.00 | 224.00 | 448.00 |
| 58.00 | 116.00 | 232.00 | 464.00 |
| 60.00 | 120.00 | 240.00 | 480.00 |
| 62.00 | 124.00 | 248.00 | 496.00 |
| 64.00 | 128.00 | 256.00 | |
| 66.00 | 132.00 | 264.00 | |
| 68.00 | 136.00 | 272.00 | |
| 70.00 | 140.00 | 280.00 | |
| 72.00 | 144.00 | 288.00 | |
| 74.00 | 148.00 | 296.00 | |

| | | |
|---|---|---|
| 76.00 | 152.00 | 304.00 |
| 78.00 | 156.00 | 312.00 |
| 80.00 | 160.00 | 320.00 |
| 82.00 | 164.00 | 328.00 |
| 84.00 | 168.00 | 336.00 |
| 86.00 | 172.00 | 344.00 |

Each global clock is delivered to the FPGA as an LVDS, differential clock. The IO input on this clock should be configured as a differential clock input (the IBUFGDS primitive).

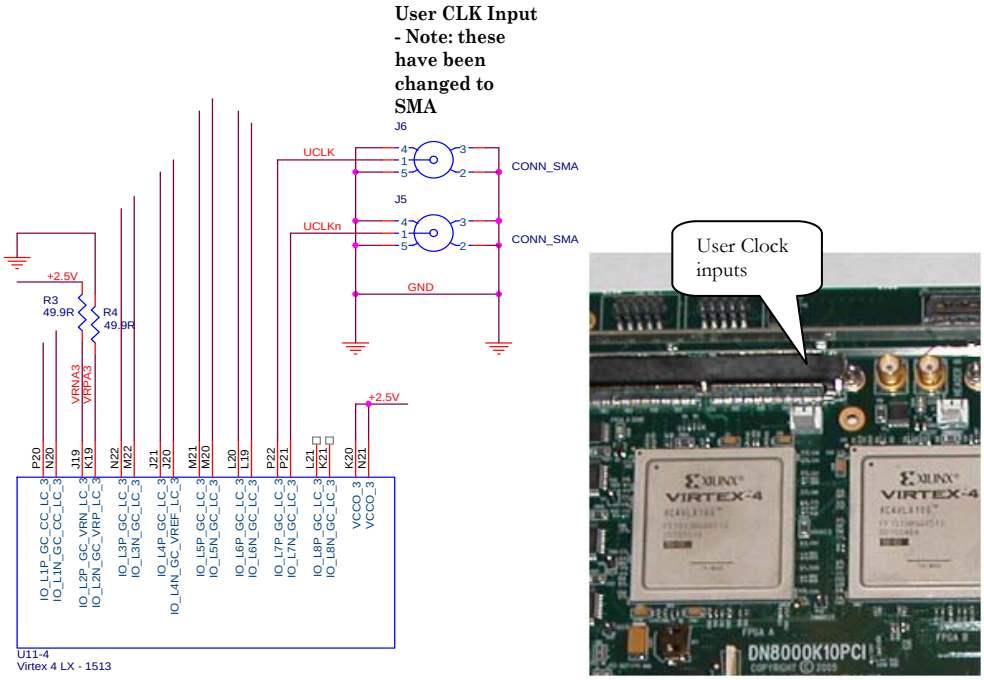
The example below shows the Verilog instantiation of this module, using the ACLK signal.

```
Wire aclk_ibufds;
IBUFGDS ACLK_IBUFG (.O(aclk_ibufg), .I(ACLK), .IB(ACLKn)) ;
```

The signal aclk_ibufds should then be fed to either a BUFG or a DCM before being used as an internal clock for FPGA logic.

## 3.2 User Clock

The DN8000K10PCI has an SMA pair reserved specifically for inputting a clock. The SMA pair is connected to a differential clock input on FPGA A (LVDS_DCI is a preferred input standard, but LVCMOS_25 will work also).



To use this clock in a synchronous design, send a copy of the clock out through the FBA (Feedback A) clock output pairs A, B and C.

For a chart of clock input pad sites on each FPGA, use the provided bioard netlist, the schematics, or the example UCF file provided with the reference design.

### 3.3 Feedback Clocks

User FPGA A and B each are capable of sourcing a clock that is distributed to all FPGAs (including back to itself). These "feedback clocks" allow the user to control a clock from inside the user design for single-stepping, multiplication/division, or distributing a clock to which only one FPGA has access (like PCI clock, a header clock, or the user clock input).

FPGA A has 6 feedback outputs, one differential pair to each Virtex 4 FPGA.

FBACLKAp/FBACLKAn, FBACLKBp/FBACLKBn, FBACLKCp/FBACLKCn

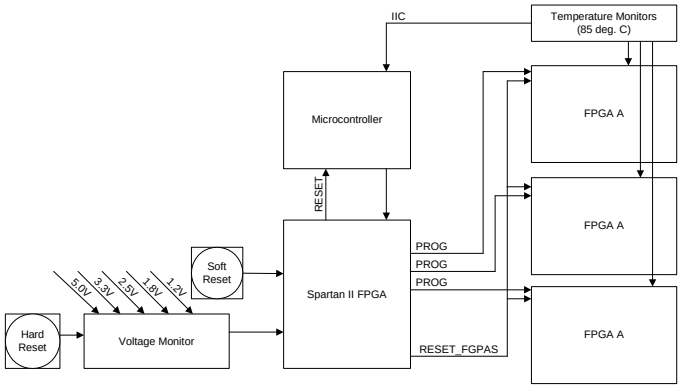
FPGA B has 6 feedback outputs, one differential pair to each Virtex 4 FPGA.

FBBCLKAp/FBBCLKAn, FBBCLKBp/FBBCLKBn, FBBCLKCp/FBBCLKCn

Clocks can also be exchanged from one FPGA to another on the source-Synchronous clock inputs. See Chapter X, Section X, FPGA interconnect.

## 4 Reset Topology

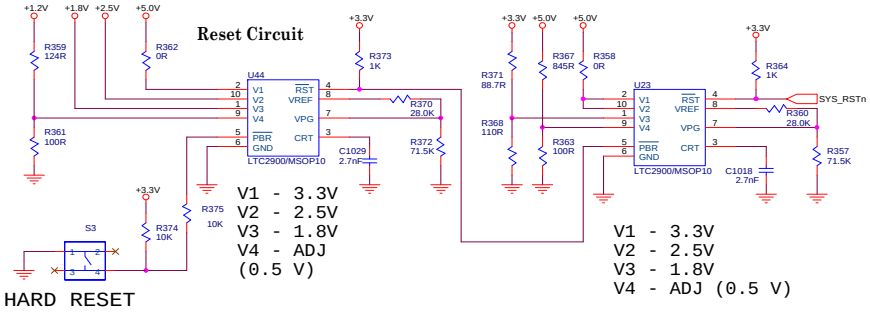
The DN8000K10PCI is protected from under voltage and over temperature by a reset circuit. When the board powers on, a voltage monitor waits until all voltages are above their minimum voltage levels, then de-asserts reset. The Spartan 2 distributes the reset signal to all FPGAs and the Microcontroller unit, so until the Spartan 2 is configured, reset remains asserted.




The user may also assert reset by pressing S3, "Hard reset" This will trigger the reset signal "SYS_RSTn" which is monitored by the Spartan FPGA. When SYS_RST is asserted, the Spartan FPGA resets the Virtex 4 FPGAs, causing them to lose their configuration data and

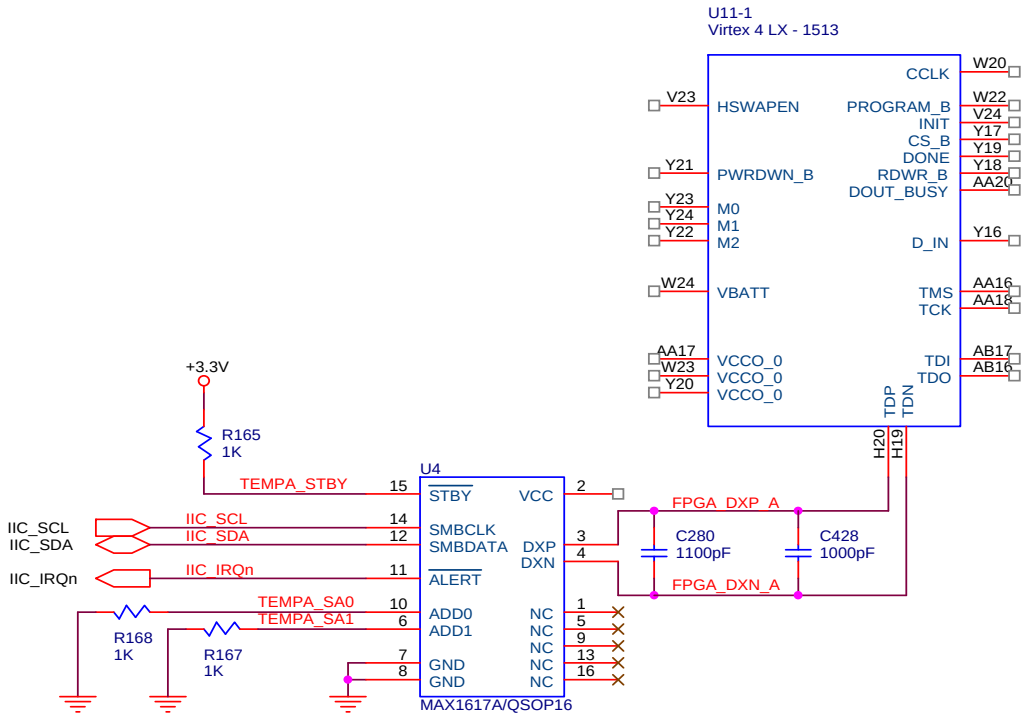
deactivate. The Spartan also causes a reset on the Microcontroller unit, which will cause the microcontroller to reload configuration instructions from the Smart Media card. USB contact will be lost with the USB host, and the DN8000K10PCI will have to re-enumerate.

There is a second button, S2 called "Soft Reset". When this button is pressed, the signal "RESET_FPGAs" is asserted. This signal is sent to the Virtex 4 FPGAs on a user IO pin, and could be used by the user design as a reset signal. This signal is also asserted to all FPGAs after any FPGA becomes configured. RESET_FPGAs is an asynchronous signal.




The above circuit shows how two LTC2900 voltage monitors are daisy chained together to monitor 5 different voltages.

Each FPGA is also connected to a temperature monitor. The Virtex 4 FPGA can easily overheat if a heatsink and fan are not used. The recommended operating temperature for the Virtex 4 is 85 degrees C. The absolute maximum temperature for operation is 125 degrees C. If at any time the junction temperature of the Virtex 4 exceeds 85 degrees, the Microcontroller will reset the FPGAs, causing them to lose their configuration data. An overheating FPGA could be the result of a mis-configuration, a clock that is set incorrectly, or an inadequate heatsink unit. The heatsink and fan assembly that comes with the DN8000K10PCI is appropriate for dissipating the amount of heat energy available through a PCI slot without the auxiliary power connector (25W total for the card). If you are operating the DN8000K10PCI at very high speeds in stand alone mode and you are causing heat overload resets, you may need to install a larger heatsink, or increase the system airflow.




This circuit shows the MAX1617 temperature monitor. The IIC bus is connected to the Cypress microcontroller.

### 4.1 User Reset

The three Virtex-4 FPGAs are also connected to a user reset signal. This signal is asserted after the FPGAs are configured, from the USB GUI controller, or using the "Soft Reset" button on the DN8000K10PCI.

| Signal Name | Pin on FPGA A | Pin on FPGA B | Pin on FPGA C |
|---|---|---|---|
| RESETn | AK19 | K21 | AG18 |

# 5 Power

The DN8000K10PCI gets is power from the 5.0V and 3.3V rails of the PCI card edge connector. It can also be operated in stand-alone mode with a 20-pin ATX power supply connector.

The PCI slot is capable of sourcing

The main rails of the DN8000K10PCI are:

- 1.2V – This is the main power supply rail used for the internal digital logic of Virtex 4 FPGAs.

- 1.8V – This is used for IO signaling and internal logic of DDR2 SDRAM memory. It is also used to supply some Gigabit optical modules, and is used as a low-power current source to supply RocketIO isolated power rails.

- 2.5V – This is used to power FPGA interconnect with low-power LVDS. It is also used as the analog power supply on the Virtex 4 FPGAs.

- 3.3V – This voltage supplies the LVDS clock distribution trees. It is also used to power the LVTTL interfaces of the Cypress microcontroller, and Quicklogic 5064 PCI bridge.

- 5.0V – This voltage is used to supply power to the 1.2, 2.5 and 1.8V switching power supplies. It also powers the FPGA cooling fans, some Gigabit optical modules, and the PCI signaling. If the PCI slot isn't providing enough power, then a Hard Drive 4-pin power cable can be connected to the board (from the same ATX power supply) to reduce the voltage droop on 5V. Please note that the board is capable of exceeding the 25W limit of the PCI connector (depending on the density of the FPGAs utilized, and the operating frequency).

The DN8000K10PCI also has these secondary rails:

- 0.9V – This voltage is used to terminate the SSTL18 signaling of the DDR2 memory module.

- RocketIO 1.2V top, 1.2V right, 1.2V bottom – These linear regulated rails are very low noise supplies for the RocketIO CML inputs and outputs. They are isolated from each other to improve the isolation of multiple RocketIO channels operating simultaneously.

- RocketIO 1.5V – This linearly regulated voltage rail supplies the internal digital logic of the RocketIOs.

- RocketIO 2.5V – this linearly regulated voltage rail supplies the internal analog circuits of the RocketIO.

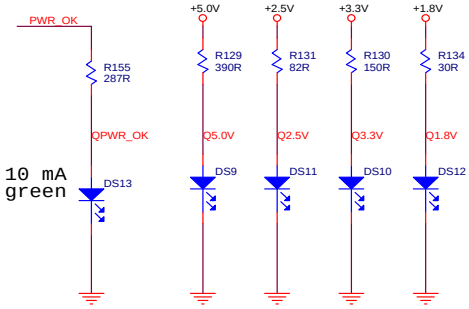
- +12V – This rail is passed directly from the PCI edge connector and ATX power connector to the Micropax expansion header. See Chapter X, Section X, Expansion Headers. Note that the fuse between +12V and the expansion headers is not installed on the board.

- -12V – This rail is passed directly from the PCI edge connector and ATX power connector to the Micropax expansion header. See Chapter X, Section X, Expansion Headers. Note that the fuse between -12V and the expansion headers is not installed on the board.

- XFP VEE5 – Power for this rail is not supplied by the DN8000K10PCI, but is required for the operation of PECL optical modules. To power this rail, you will need to connect an external power connector to the board from a low-noise voltage supply.

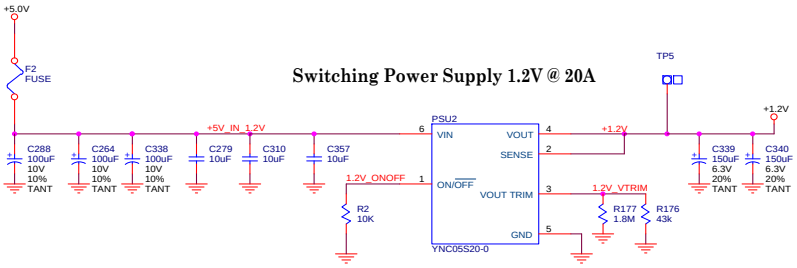
There are test points for measuring the voltage levels of each rail near the top left of the DN8000K10PCI. Each rail is monitored by a voltage monitor circuit, and will cause a reset if any of the primary supplies drop 5% or more below their set points.

There are also LEDs next to each test point to indicate the presence of each voltage rail. These LEDs do not indicate that a rail is within 5% of its set point, only that the rail is present and above 1.6V. A power OK led shows the status of the ATX power supply's PWR_OK signal. If this LED is lit, then +5.0V and +3.3V (and +12V –12V) are within 5% of their set points.




## 5.1 Switching power supplies

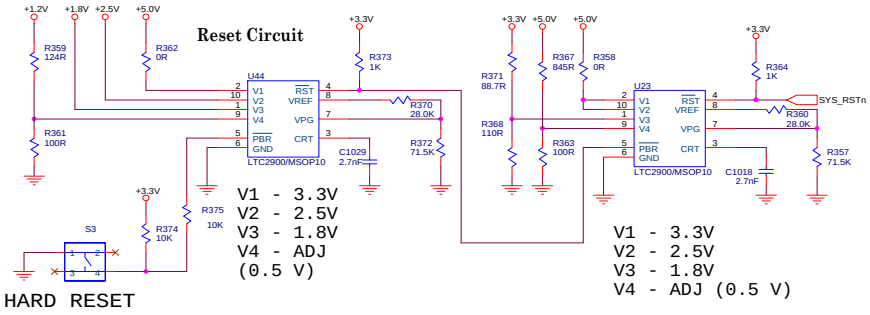
The main power rails for the Virtex 4 FPGAs are produced on board with three 20A switching power supplies, one for each of 1.8V, 2.5V, and 1.2V.




The DN8000K10PCI is shipped with a fun mounted above the power supplies to help keep them cool. If you need to remove this fan, the DN8000K10PCI will function properly without it, but be careful not to touch the power supplies with your fingers because they will burn!

Each power supply is protected with a 15A fuse on the inputs. If you need to operate the DN8000K10PCI with more than 15A of current for a power supply, you can change this fuse, but you need to find a heatsink solution for keeping the Virtex 4 FPGAs cool. The heatsink and fan provided are appropriate for a power consumption of about 10-15W per FPGA.

Each of the primary power rails (5.0, 3.3, 2.5, 1.8, 1.2) is monitored for under voltage. If the voltage monitor circuit detects a low voltage, it will hold the board in reset until the supply is back within 5% of its set point. See section X, Reset Circuit for information on reset.

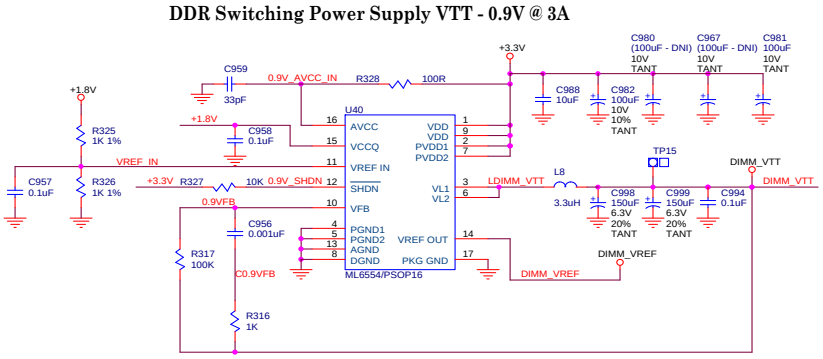



## 5.2 Secondary Power Supplies

The secondary power supplies are derived from a primary supply.

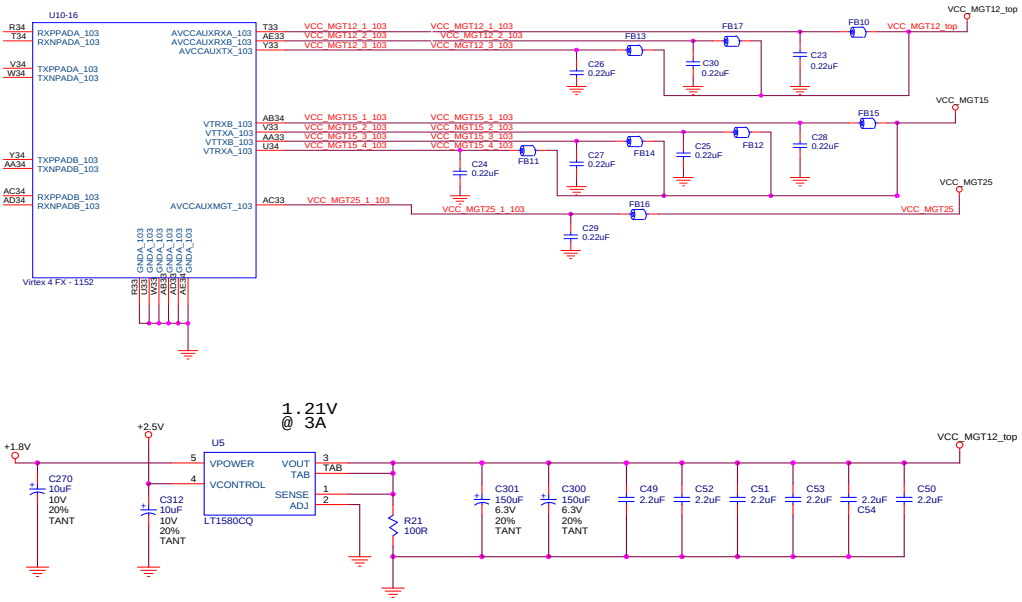
### 5.2.1 DDR2 Termination Power

DDR2 memory modules use the SSTL18 signaling standard. Properly terminating SSTL18 requires a termination power supply of 0.9V. Since as much as 1.6 Amps of termination current are needed, a switching power supply is required.




The ML6554 produces up to 3A of the required 0.9V termination power rail along with a stable 0.9V reference voltage supply.

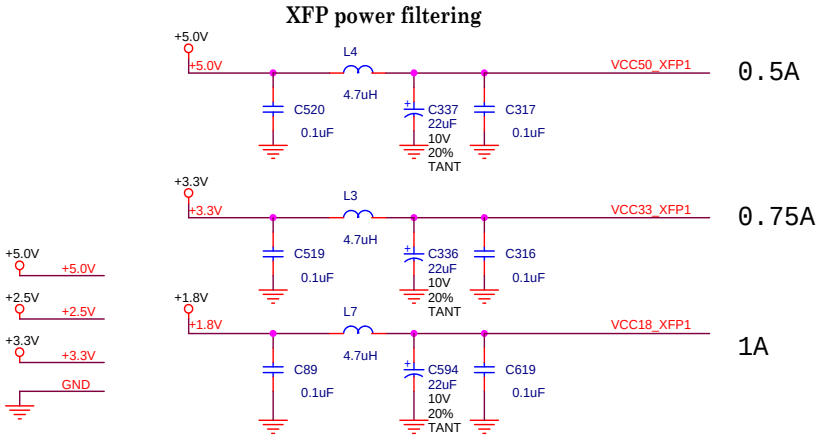
### 5.2.2 RocketIO power



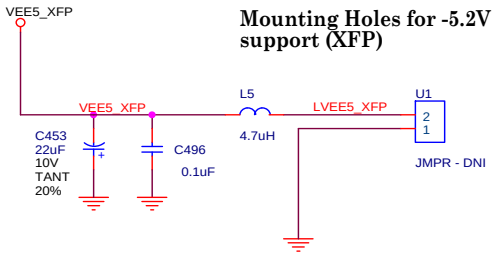

Five linear rails

### 5.2.3 Optical Module Power

Optional optical modules have a variety of power supply requirements, most of which are met by the DN8000K10PCI.

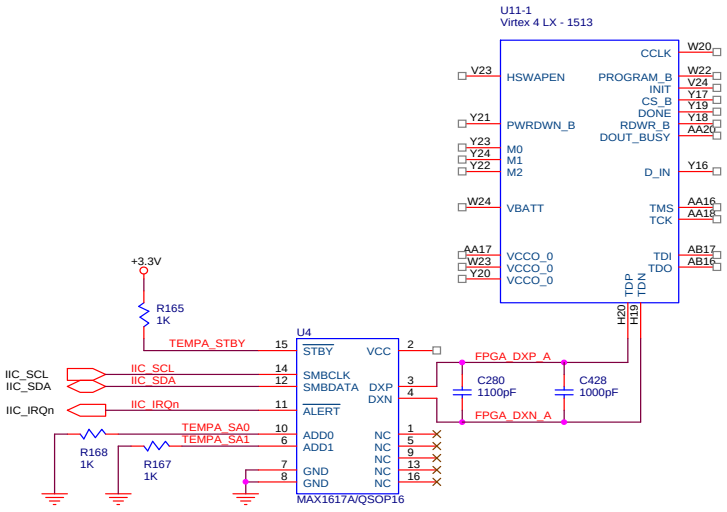



Since the DN8000K10PCI has no negative voltage supply, it cannot generate the –5.2V required to supply ECL-based optical transceiver modules. An auxiliary power connector is supplied to connect to an external voltage supply if ECL signaling is required.

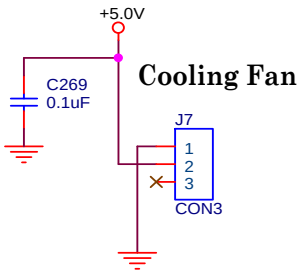



## 5.3 Heat dissipation

Virtex 4 FPGAs are capable of drawing incredible amounts of current from their 1.2V and 2.5V power supplies. According to Xilinx online power estimator tool, a fully utilized FPGA running at 300Mhz can draw more than 30W of power. With this much power used in each FPGA, the DN8000K10PCI can dissipate 75 or more Watts of heat. For all but the most trivial designs, a heatsink must be used with the Virtex 4 FPGA. The DN8000K10PCI comes with a forced air heatsink rated at 2 degrees per Watt. Since the maximum operating junction temperature of a Virtex 4 FPGA is 85 degrees, assuming an ambient temperature of 50 degrees (the inside of your computer case) the most amount of energy dissipated by the FPGA using the standard fan is 85 – 30 / 2 = 27.5W. This should be sufficient for most applications. If you intend to operate the Virtex 4 FPGA at very high speeds, or are getting overheating issues with your design, you will need to install a larger heatsink.




Above: The FPGA temperature monitor circuit. The MAX1617's IIC bus is connected to the Cypress MCU.



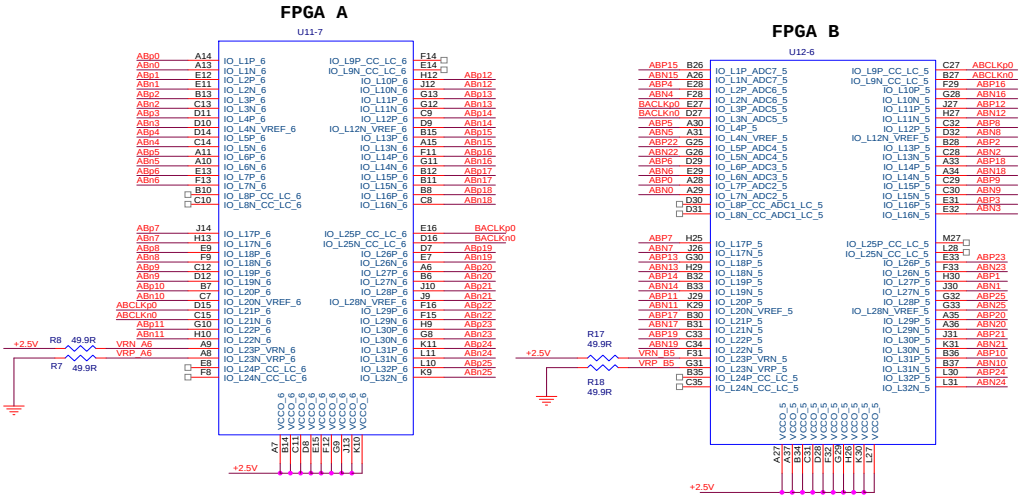

Above: Cooling fan power connector.

# 6 FPGA interconnect

The DN8000K10PCI was designed to maximize the amount of interconnect between the two primary Virtex 4 FPGAs A and B. This interconnect was routed as tightly coupled differential LVDS to provide the best immunity to power supply and crosstalk noise so that your interconnect can operate at the full switching speed of the output buffers.

## 6.1 FPGA interconnect

Following Xilinx recommendations, the interconnect on the DN8000K10PCI was designed to operate at 1Gb/s for every LVDS pair. (Note 1Gb/s operation requires the fasted speed-grade part, LX200 –12) In order to achieve such breakneck speeds, you will need to operate the busses of signals using a source-synchronous clocking scheme. The interconnect signals on the DN8000K10PCI have been optimized to operate in "lanes" There are 7 lanes between FPGAs A and B, three between B and C and two between FPGAs A and C. Each lane has a differential LVDS source-synchronous clock in each direction.



Clocking incoming data at high speeds required the used of the each input's delay buffer to align each bit. The incoming clock needs to be adjusted and used to clock the inputs within its lane. This process can be automated by the use of the new Virtex 4 feature IDELAYCTL.

For detailed description of the required user design to achieve 1Gbs operation, see Xilinx Application note XAPP704, "High Speed SDR LVDS Transceiver".'

Synchronous clocking and single-ended signaling are still possible on the DN8000K10PCI, you are not required to use high speed serial design techniques. Single ended interconnect is recommended for signaling below 133Mhz. Because of the DN8000K10PCI's excellent low-skew clocking network, global synchronous clocking should work fine for your interconnect at speeds lower than 300Mhz. The source synchronous clock signals can also be used as single ended or differential interconnect, or to forward clocks from one FPGA to another.

The total interconnect counts between FPGAs

- A-B 378
- B – C 154
- A – C 112

The following signals should be used for the source-synchronous clocking requirements of the serdes inter-fpga interface. Each bank of interconnect requires a single source-synchronous clock in each direction.

FPGA A to B signals in bank 0 should be clocked using ABCLKp0 and ABCLKn0. Signals in the same bank in the opposite direction should use BACLKp0 and BACLKn0 as a source-synchronous clock. These clock signals should be driven as LVDS signals and received using the DIFF_TERM attribute.

| AAA_Signal Name | pin on FPGA A | pin on FPGA C |
|---|---|---|
| FROM A TO C | | |
| ACCLKN0 | C2 | AE31 |
| ACCLKN1 | U2 | AK21 |
| ACCLKP0 | C3 | AE32 |
| ACCLKP1 | U3 | AL21 |
| FROM C TO A | | |
| CACLKN0 | G1 | AJ30 |
| CACLKN1 | M2 | AL28 |
| CACLKP0 | F1 | AH30 |
| CACLKP1 | M3 | AK28 |

| AAA_Name | pin on FPGA B | pin on FPGA C |
|---|---|---|

| FROM B TO C | | |
|---|---|---|
| BCCLKN0 | G37 | E29 |
| BCCLKN1 | AA39 | G20 |
| BCCLKN2 | W30 | D15 |
| BCCLKP0 | G36 | F29 |
| BCCLKP1 | Y39 | F20 |
| BCCLKP2 | W29 | D16 |
| FROM C TO B | | |
| CBCLKN0 | D39 | C32 |
| CBCLKN1 | W39 | E21 |
| CBCLKN2 | AJ37 | H7 |
| CBCLKP0 | C39 | D32 |
| CBCLKP1 | V39 | D21 |
| CBCLKP2 | AJ36 | G7 |

| AA Clock A to B | pin on FPGA A | pin on FPGA B |
|---|---|---|
| FROM A TO B | | |
| ABCLKN0 | C15 | C27 |
| ABCLKN1 | G26 | D16 |
| ABCLKN2 | L39 | G1 |
| ABCLKN3 | U38 | M2 |
| ABCLKN4 | AG38 | AE2 |
| ABCLKN5 | AU37 | AU1 |
| ABCLKN6 | AW27 | AV12 |
| ABCLKP0 | D15 | B27 |
| ABCLKP1 | G25 | E16 |
| ABCLKP2 | K39 | F1 |
| ABCLKP3 | T38 | M3 |
| ABCLKP4 | AG37 | AE3 |
| ABCLKP5 | AV37 | AU2 |
| ABCLKP6 | AV27 | AW12 |
| FROM B TO A | | |
| BACLKN0 | D16 | D27 |
| BACLKN1 | B27 | C15 |
| BACLKN2 | D39 | A4 |
| BACLKN3 | W39 | M1 |
| BACLKN4 | AJ37 | AD4 |
| BACLKN5 | AT38 | AT3 |
| BACLKN6 | AT25 | AR13 |
| BACLKP0 | E16 | E27 |
| BACLKP1 | C27 | D15 |
| BACLKP2 | C39 | A5 |
| BACLKP3 | V39 | L1 |

| | | |
|---|---|---|
| BACLKP4 | AJ36 | AD5 |
| BACLKP5 | AU38 | AU3 |
| BACLKP6 | AT24 | AT13 |

## 6.2 Main Bus

There is a 40-signal wide bus connecting all three Virtex 4 FPGAs. This bus is used by the Dini Group reference design to communicate over USB and PCI. If you require PCI communication, you will likely communicate directly from FPGA A to the Quicklogic 5064, and not need the main bus. If you need to communicate over USB from your user design, you may want to copy the main bus interface module out of the Dini Group reference design, so you can use the existing USB software.

If the main bus is not required, these 40 signals can be used by the user design for inter-FPGA communication. In the USB Controller application that came with your User CD, there is a button near the top of the screen labeled "Enable USB<->FPGA communication" As long as this button is present, the USB Controller program will not cause the main bus signals to be driven from the Spartan 2 FPGA. If the button text instead reads "Disable USB<->FPGA communication", then the USB controller can at any time cause signals to be driven over these 40 signals, possibly disrupting any user use of these signals.

Also note that the data path that the Spartan and MCU use to communicate over USB is shared by the SmartMedia card. If you are running the Reference design tests, or have copied the reference design in order to communicate over USB, removing or inserting a SmartMedia card in the media card slot may interfere with USB Communication.

## 6.3 DCI

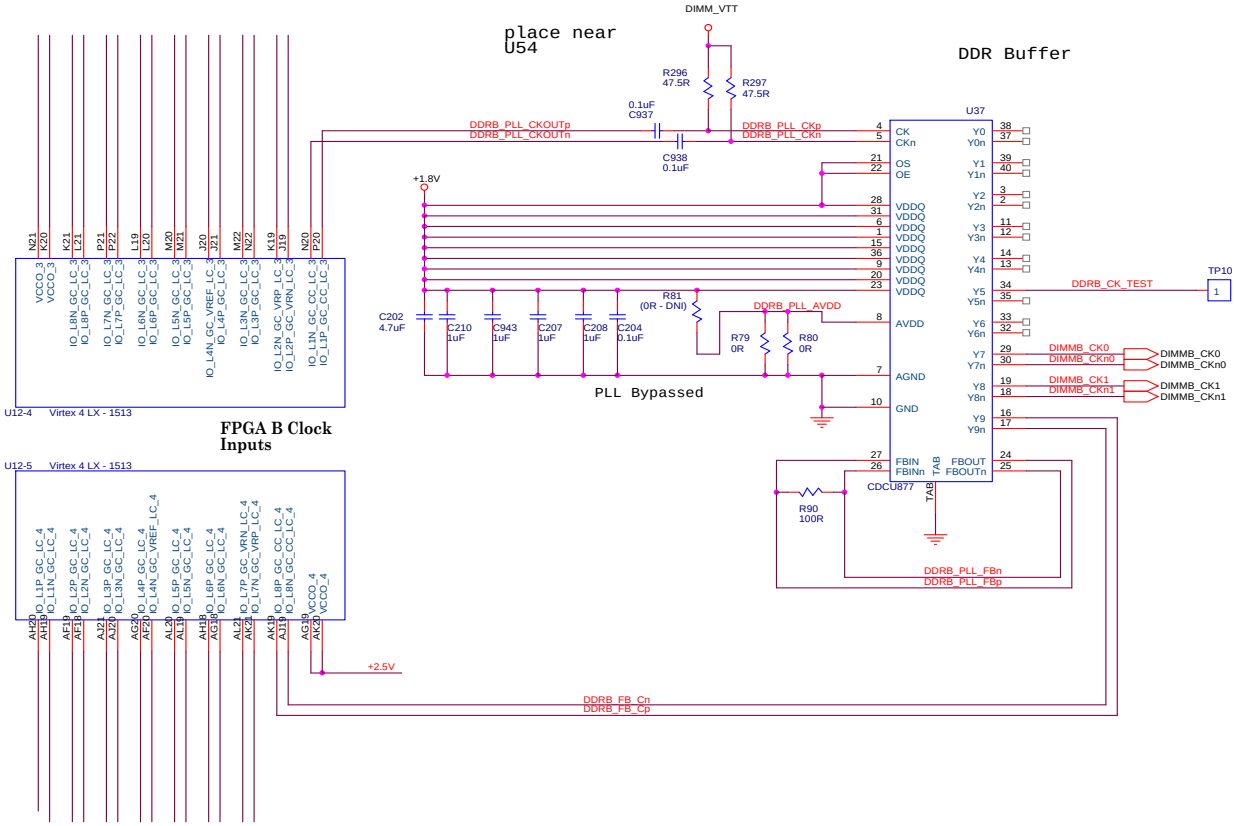
The Xilinx SelectIO technology guarantees proper signal termination of FPGA interconnect. All DCI IOs on the Virtex 4 FPGA have been properly connected to 50-Ohm reference resistors. These IOs (VRN, VRP) must not be used by the user design for DCI to function properly. When signaling between FPGAs, use the LVCMOS25_DCI signal standard for maximum performance.

Remember to achieve the highest interconnect throughput when operating the interconnect synchronously (not using the serdes), your IO clock must be deskewed using a DCM, and you must use IO flip-flops on all inputs and outputs.

# 7 Memory interface

There are two standard 200-pin DDR2 SODIMM module sockets on the DN8000K10PCI. These sockets are supplied with 1.8V power and keyed for use with DDR2 SDRAMs. One socket is connected to FPGA B and the other is connected to FPGA C.

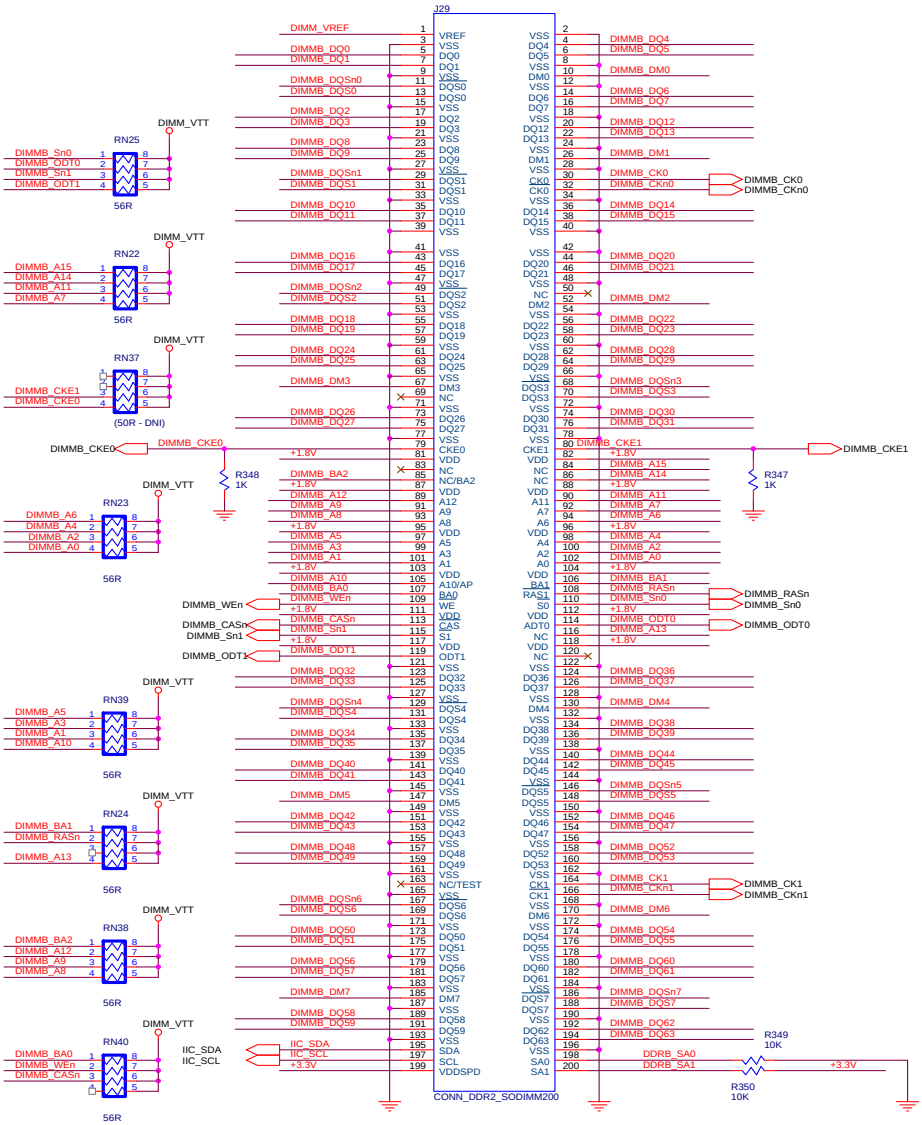
## 7.1 Clocking




A clock is provided to the SODIMM modules from the FPGA (signal DDR_PLL_CLKOUT). This is buffered through a CDCU877 SSTL clock buffer and sent to the SODIMM modules (Signal DIMM_CK0, DIMM_CK1). There is no length relationship between the clock signals to the DIMMS and the data signals to the DIMMs. The user should use the DDR_FB_C signal with a DCM to synchronize the FPGA clock to the clock received by the DIMM module.

## 7.2 Serial presence detect.

The EEPROM on the SODIMM is accessible by PCI, USB, or configuration UART.




The DIMM_CKE signals are not end-terminated. These signals should not be operated at full-frequency.

# 8 Headers

There are two daughter card headers on the DN8000K10PCI; one attached to FPGA A (Header A), and one attached to FPGA B (Header B). Header A contains 135 user IOs designed to operate as 67 differential pairs. Header B has 154 user IOs that can be used as 77 differential pairs.

The signals RESET_FPGAs is driven by the Spartan Configuration FPGA. This signal is the same as the RESET_FPGAs driven to FPGAs A B and C.

PDETECTA and PDETECTB are single-ended signal with an external pull up resistor. The daughter card can ground these signals to indicate the daughter card's presence.

The HAp/nCC and HBp/nCC signals are connected to global clock input pins on the FPGAs. These can be used as differential clock inputs from the daughter card headers to the FPGAs. They can also be used as outputs.

The ACLK and BCLK signals are copies of the DN8000K10PCI global differential clocks ACLK and BCLK. The signals are synchronized at the daughter card connector with the ACLK and BCLK signals at the pins of the FPGA.

The signals HAp*XX*/HAn*XX* and HBp*XX*/HBn*XX* are connected to Virtex 4 IOs. Generally, these signals are LVCMOS25 outputs, LVCMOS25 inputs, or LVDS inputs. Some of these signals are connected to BANK 1 of FPGA A and B. On the DN8000K10PCI, these two banks can be set to operate at +3.3V. This allows the use of LVCMOS33 outputs. (LVDS inputs are still OK). The table below contains the list of "BANK 1" IO sites on a Virtex 4 FPGA.

Table 2 FPGA A and B bank 1 pins

| | | | | | |
|---|---|---|---|---|---|
| F26 | F25 | K16 | L16 | E26 | J16 |
| M25 | G16 | T23 | A16 | D26 | H15 |
| N24 | G15 | R22 | B16 | D20 | E19 |
| D21 | C18 | D22 | G20 | J22 | G22 |
| F21 | N18 | G23 | M18 | E22 | H17 |
| B23 | E18 | C20 | D19 | E21 | C19 |
| C22 | F19 | H22 | T19 | P19 | H23 |
| L18 | F23 | G18 | C23 | F18 | T20 |
| L24 | N17 | J17 | D25 | D17 | |
| M17 | K24 | K23 | | | |
| C17 | R17 | C25 | B25 | A25 | E17 |
| A18 | E24 | P17 | F24 | B17 | J24 |
| U20 | A24 | B18 | D24 | C24 | K17 |

| | | | | | |
|---|---|---|---|---|---|
| U18 | A23 | T18 | R18 | N23 | M23 |

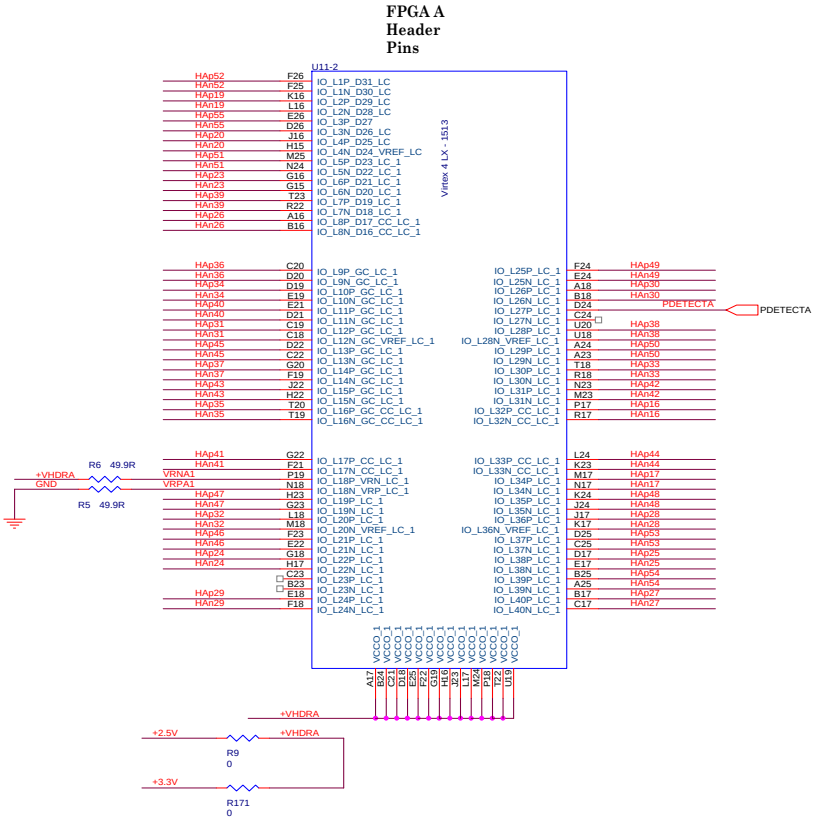
Header B has more signals than Header A. A daughter card designed to work with header A will work with header B.

## 8.1 3000K10 Compatibility

The DN8000K10PCI headers use pinout similar to that on the DN3000K10. A compatibility chart with the DN3000K10SD and Mictor daughter cards is given on the user CD in the daughter card directory. Also see the customer netlist in the schematics directory. The +1.5V power supplies, MBCLKA-F are not present.

## 8.2 FPGA Connection

On the DN8000K10PCI, all header signals are connected to "LC" pins on the Virtex 4 FPGA. See the Virtex 4 User's Guide for detail about these signals. The main result of this is that the headers on the DN8000K10PCI may not be used with the Virtex 4's current-mode LVDS drivers. Virtex 4 LVDS receivers may still be used. Differential outputs can be acheieved using complementary logic and output flip-flops.




On both Header A and Header B, there is a bank that is dedicated entirely to the Headers. For details about Virtex 4 IO banks, see the Virtex 4 user guide. This bank can be used for standards

requiring a threshold voltage reference, such as SSTL. You can also use this bank for source-synchronous clocking.

| Header Pin Number | Signal Name | FPGA A Pin | Signal Name | FPGA B Pin |
|---|---|---|---|---|
| 1 | NC/12V | | NC | |
| 2 | GND | | GND | |
| 3 | +2.5V | | +2.5V | |
| 4 | +5.0V | | +5.0V | |
| 5 | +2.5V | | +2.5V | |
| 6 | +5.0V | | +5.0V | |
| 7 | NC | | NC | |
| 8 | GND | | GND | |
| 9 | +3.3V | | +3.3V | |
| 10 | NC | | NC | |
| 11 | GND | | GND | |
| 12 | BCLKHAP | | BCLKp | |
| 13 | BCLKHAN | | BCLKn | |
| 14 | NC | | HBP67 | AF34 |
| 15 | NC | | HBN67 | AF35 |
| 16 | NC | | HBP73 | AG23 |
| 17 | NC | | HBN73 | AH23 |
| 18 | NC | | HBP71 | AG22 |
| 19 | NC | | HBN71 | AF21 |
| 20 | NC | | HBP69 | AE22 |
| 21 | NC | | HBN69 | AD21 |
| 22 | GND | | GND | |
| 23 | NC | | HBP75 | AH17 |
| 24 | NC | | HBN75 | AG17 |
| 25 | HAP65 | AB33 | HBP65 | W34 |
| 26 | HAN65 | AA33 | HBN65 | W35 |
| 27 | HAP63 | B35 | HBP63 | R36 |
| 28 | HAN63 | C35 | HBN63 | T36 |
| 29 | HAP61 | J32 | HBP61 | R28 |
| 30 | HAN61 | K32 | HBN61 | R29 |
| 31 | HAP59 | D30 | HBP59 | J32 |
| 32 | HAN59 | D31 | HBN59 | K32 |
| 33 | GND | | GND | |
| 34 | HAP57 | R28 | HBP57 | D30 |
| 35 | HAN57 | R29 | HBN57 | D31 |
| 36 | HAP55 | E26 | HBP55 | N23 |
| 37 | HAN55 | D26 | HBN55 | M23 |
| 38 | HAP53 | D25 | HBP53 | T23 |
| 39 | HAN53 | C25 | HBN53 | R22 |
| 40 | HAP51 | M25 | HBP51 | T20 |

| | | | | |
|---|---|---|---|---|
| 41 | HAN51 | N24 | HBN51 | T19 |
| 42 | HAP49 | F24 | HBP49 | T18 |
| 43 | HAN49 | E24 | HBN49 | R18 |
| 44 | GND | | GND | |
| 45 | HAP47 | H23 | HBP47 | P17 |
| 46 | HAN47 | G23 | HBN47 | R17 |
| 47 | HAP45 | D22 | HBP45 | M17 |
| 48 | HAN45 | C22 | HBN45 | N17 |
| 49 | HAP43 | J22 | HBP43 | F26 |
| 50 | HAN43 | H22 | HBN43 | F25 |
| 51 | HAP41 | G22 | HBP41 | F24 |
| 52 | HAN41 | F21 | HBN41 | E24 |
| 53 | HAP39 | T23 | HBP39 | F23 |
| 54 | HAN39 | R22 | HBN39 | E22 |
| 55 | GND | | GND | |
| 56 | HAP37 | G20 | HBP37 | G22 |
| 57 | HAN37 | F19 | HBN37 | F21 |
| 58 | HAP35 | T20 | HBP35 | G20 |
| 59 | HAN35 | T19 | HBN35 | F19 |
| 60 | HAP33 | T18 | HBP33 | D19 |
| 61 | HAN33 | R18 | HBN33 | E19 |
| 62 | HAP31 | C19 | HBP31 | E18 |
| 63 | HAN31 | C18 | HBN31 | F18 |
| 64 | HAP29 | E18 | HBP29 | D17 |
| 65 | HAN29 | F18 | HBN29 | E17 |
| 66 | GND | | GND | |
| 67 | HAP27 | B17 | HBP27 | G16 |
| 68 | HAN27 | C17 | HBN27 | G15 |
| 69 | HAP25 | D17 | HBP25 | E26 |
| 70 | HAN25 | E17 | HBN25 | D26 |
| 71 | HAP23 | G16 | HBP23 | D25 |
| 72 | HAN23 | G15 | HBN23 | C25 |
| 73 | HAP21 | AB15 | HBP21 | D24 |
| 74 | HAN21 | AC14 | HBN21 | C24 |
| 75 | HAP19 | K16 | HBP19 | C23 |
| 76 | HAN19 | L16 | HBN19 | B23 |
| 77 | GND | | GND | |
| 78 | HAP17 | M17 | HBP17 | D22 |
| 79 | HAN17 | N17 | HBN17 | C22 |
| 80 | HAP15 | E8 | HBP15 | E21 |
| 81 | HAN15 | F8 | HBN15 | D21 |
| 82 | HAP13 | G7 | HBP13 | C20 |
| 83 | HAN13 | G6 | HBN13 | D20 |
| 84 | HAP11 | AA14 | HBP11 | C19 |
| 85 | HAN11 | AA13 | HBN11 | C18 |

| | | | | |
|---|---|---|---|---|
| 86 | HAP9 | AB13 | HBP9 | B17 |
| 87 | HAN9 | AC13 | HBN9 | C17 |
| 88 | GND | | GND | |
| 89 | HAP7 | T1 | HBP7 | U5 |
| 90 | HAN7 | U1 | HBN7 | V5 |
| 91 | HAP5 | AK29 | HBP5 | AJ4 |
| 92 | HAN5 | AJ29 | HBN5 | AH4 |
| 93 | NC | | NC | |
| 94 | HAP3 | AP27 | HBP3 | AT1 |
| 95 | HAN3 | AR27 | HBN3 | AR1 |
| 96 | RESETN | | RESETN | |
| 97 | HAP1 | AJ34 | HBP1 | T1 |
| 98 | HAN1 | AH34 | HBN1 | U1 |
| 99 | GND | | GND | |
| 100 | NC/-12V | | NC | |
| 101 | GND | | GND | |
| 102 | HACLKA | | HBCLKA | |
| 103 | NC | | NC | |
| 104 | GND | | GND | |
| 105 | +3.3V | | +3.3V | |
| 106 | NC | | NC | |
| 107 | GND | | GND | |
| 108 | GND | | GND | |
| 109 | GND | | GND | |
| 110 | GND | | GND | |
| 111 | ACLKHAP | | ACLKp | |
| 112 | ACLKHAN | | ACLKn | |
| 113 | PDETECTA | | PDETECTB | |
| 114 | HAP76 (HACC) | P20 | HBP76 (HBCC) | AH18 |
| 115 | HAN76 (HACC) | N20 | HBN76 (HBCC) | AG18 |
| 116 | NC | | HBP66 | AB33 |
| 117 | NC | | HBN66 | AA33 |
| 118 | GND | | GND | |
| 119 | NC | | HBP72 | AE23 |
| 120 | NC | | HBN72 | AF23 |
| 121 | NC | | HBP74 | AH22 |
| 122 | NC | | HBN74 | AJ22 |
| 123 | NC | | HBP68 | AM23 |
| 124 | NC | | HBN68 | AN22 |
| 125 | HAP66 | AC33 | HBP70 | AE18 |
| 126 | HAN66 | AC34 | HBN70 | AD17 |
| 127 | HAP64 | R36 | HBP64 | T31 |
| 128 | HAN64 | T36 | HBN64 | U30 |
| 129 | GND | | GND | |
| 130 | HAP62 | W34 | HBP62 | M33 |

| | | | | |
|---|---|---|---|---|
| 131 | HAN62 | W35 | HBN62 | N32 |
| 132 | HAP60 | M33 | HBP60 | B35 |
| 133 | HAN60 | N32 | HBN60 | C35 |
| 134 | HAP58 | T31 | HBP58 | M27 |
| 135 | HAN58 | U30 | HBN58 | L28 |
| 136 | HAP56 | M27 | HBP56 | M25 |
| 137 | HAN56 | L28 | HBN56 | N24 |
| 138 | HAP54 | B25 | HBP52 | L24 |
| 139 | HAN54 | A25 | HBN52 | K23 |
| 140 | GND | | GND | |
| 141 | HAP52 | F26 | HBP54 | K24 |
| 142 | HAN52 | F25 | HBN54 | J24 |
| 143 | HAP50 | A24 | HBP50 | H23 |
| 144 | HAN50 | A23 | HBN50 | G23 |
| 145 | HAP48 | K24 | HBP48 | J22 |
| 146 | HAN48 | J24 | HBN48 | H22 |
| 147 | HAP46 | F23 | HBP46 | L18 |
| 148 | HAN46 | E22 | HBN46 | M18 |
| 149 | HAP44 | L24 | HBP44 | J17 |
| 150 | HAN44 | K23 | HBN44 | K17 |
| 151 | GND | | GND | |
| 152 | HAP42 | N23 | HBP42 | G18 |
| 153 | HAN42 | M23 | HBN42 | H17 |
| 154 | HAP40 | E21 | HBP40 | K16 |
| 155 | HAN40 | D21 | HBN40 | L16 |
| 156 | HAP38 | U20 | HBP38 | J16 |
| 157 | HAN38 | U18 | HBN38 | H15 |
| 158 | HAP36 | C20 | HBP36 | B25 |
| 159 | HAN36 | D20 | HBN36 | A25 |
| 160 | HAP34 | D19 | HBP34 | A24 |
| 161 | HAN34 | E19 | HBN34 | A23 |
| 162 | GND | | GND | |
| 163 | HAP32 | L18 | HBP32 | A18 |
| 164 | HAN32 | M18 | HBN32 | B18 |
| 165 | HAP30 | A18 | HBP30 | A16 |
| 166 | HAN30 | B18 | HBN30 | B16 |
| 167 | HAP28 | J17 | HBP28 | F14 |
| 168 | HAN28 | K17 | HBN28 | E14 |
| 169 | HAP26 | A16 | HBP26 | AA14 |
| 170 | HAN26 | B16 | HBN26 | AA13 |
| 171 | HAP24 | G18 | HBP24 | AC12 |
| 172 | HAN24 | H17 | HBN24 | AD11 |
| 173 | GND | | GND | |
| 174 | HAP22 | F14 | HBP22 | P11 |
| 175 | HAN22 | E14 | HBN22 | R11 |

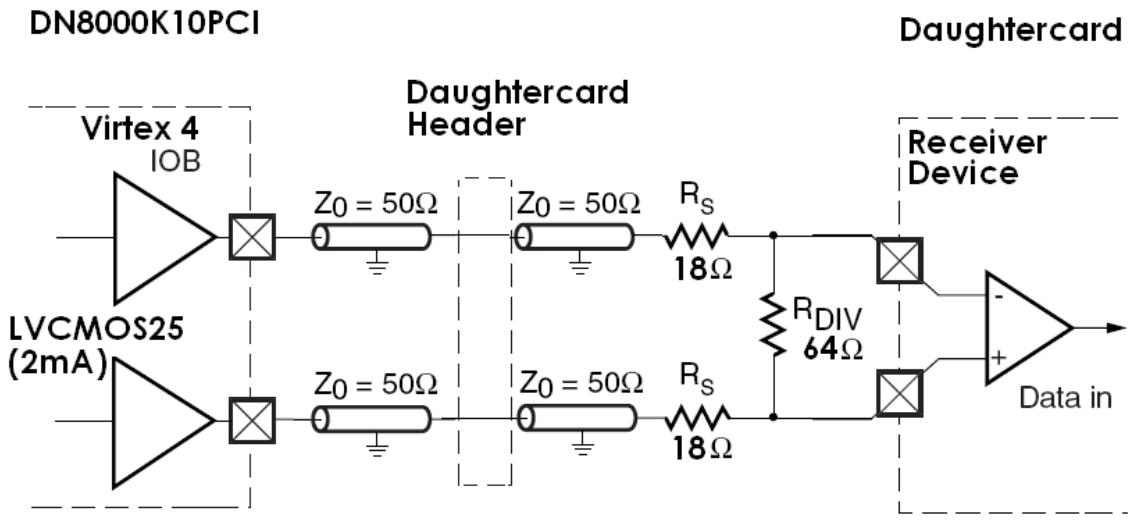
| | | | | |
|---|---|---|---|---|
| 176 | HAP20 | J16 | HBP20 | B10 |
| 177 | HAN20 | H15 | HBN20 | C10 |
| 178 | HAP18 | B10 | HBP18 | AL11 |
| 179 | HAN18 | C10 | HBN18 | AK11 |
| 180 | HAP16 | P17 | HBP16 | AV8 |
| 181 | HAN16 | R17 | HBN16 | AV7 |
| 182 | HAP14 | H7 | HBP14 | E8 |
| 183 | HAN14 | J7 | HBN14 | F8 |
| 184 | GND | | GND | |
| 185 | HAP12 | P11 | HBP12 | G7 |
| 186 | HAN12 | R11 | HBN12 | G6 |
| 187 | HAP10 | N5 | HBP10 | H7 |
| 188 | HAN10 | N4 | HBN10 | J7 |
| 189 | HAP8 | U5 | HBP8 | AJ6 |
| 190 | HAN8 | V5 | HBN8 | AJ5 |
| 191 | HAP6 | AC10 | HBP6 | N5 |
| 192 | HAN6 | AD9 | HBN6 | N4 |
| 193 | HAP4 | AL28 | HBP4 | AV4 |
| 194 | HAN4 | AM28 | HBN4 | AV3 |
| 195 | GND | | GND | |
| 196 | HAP2 | AF34 | HBP2 | Y1 |
| 197 | HAN2 | AF35 | HBN2 | AA1 |
| 198 | GND | | GND | |
| 199 | HAP0 | AU35 | HBP0 | AT14 |
| 200 | HAN0 | AU36 | HBN0 | AR14 |

## 8.3 Getting LVDS on the header

Since all of the FPGA pins connected to header A and header B are "LC" pins, the Virtex-4 "LVDS" standard cannot be used as outputs. LVDS signaling is still possible.

On inputs, use the LVDS25 standard.

On outputs, use two LVCMOS25 DRIVE=2 SLEW=FAST. On the daughtercard, terminate the signal with the following circuit




This circuit terminates the differential signal at the destination with 100-Ohms differential impedance. It also divides the voltage produced by the LVCMOS25 buffers down to a <400mV differential voltage. The power requirement for this IO remains fixed at 6.2mA regardless of frequency of operation.

## 8.4 IO Power

The IOs connected to the headers on the Virtex 4 FPGAs are powered with a +2.5V power rail.

## 8.5 Physical

Micropax part number FCI 91294-003

The standard Dini Group mounting hole location for all 200-pin Micropax connections is (430 mils)

## 8.6 Daughter card Power

Power is supplied to the daughter card though dedicated power supply pins. The maximum allowed current for each of the daughter card supplies is

5.0V – 1A
3.3V – 1A
2.5V – 1A
12V – 250mA
-12V – 250mA

The 12V and –12V supplies are by default disconnected by removing the series jumper resistors R413, R412, R411, R414. This help prevent accidental damage due to careless probing. The 12V and –12V supplies may be able to source as much as 0.5A of current if the current can be supplied by the host PC.

## 8.7 The Mictor

There is a Mictor connected designed to be used with an agilent logic analyzer. Riscwatch power PC debugger can also be used over this connection.

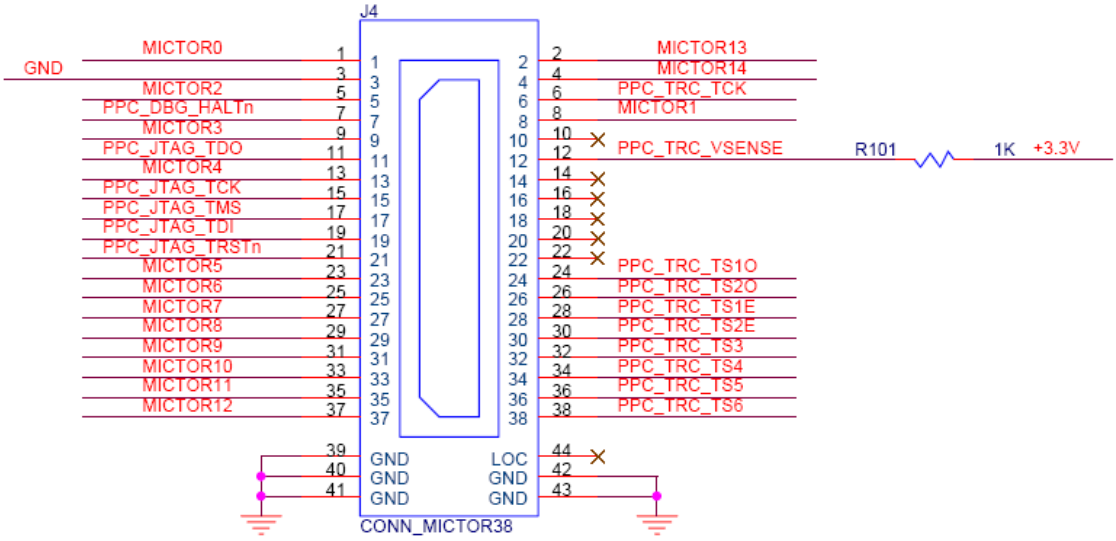



Figure 31 Mictor Header

| Signal Name | Pin on FPGA C | RISC Watch signal |
|---|---|---|
| MICTOR.1 | AM10 | |
| MICTOR.2 | AJ15 | |
| MICTOR.5 | AF13 | |
| MICTOR.6 | AF9 | PPC_TRC_TCK |
| MICTOR.7 | AG13 | PPC_DBG_HALTn |
| MICTOR.9 | AE13 | |
| MICTOR.11 | AH13 | PPC_JTAG_TDO |
| MICTOR.13 | AH9 | |
| MICTOR.15 | AJ12 | PPC_JTAG_TCK |
| MICTOR.17 | AK12 | PPC_JTAG_TMS |
| MICTOR.19 | AF11 | PPC_JTAG_TDI |
| MICTOR.21 | AG11 | PPC_JTAG_TRSTn |
| MICTOR.23 | AJ9 | |
| MICTOR.24 | AG12 | PPC_TRC_TS1O |
| MICTOR.25 | AF14 | |
| MICTOR.26 | AH12 | PPC_TRC_TS2O |
| MICTOR.27 | AC12 | |
| MICTOR.28 | AM13 | PPC_TRC_TS1E |
| MICTOR.29 | AH7 | |
| MICTOR.30 | AM12 | PPC_TRC_TS2E |

| | | |
|---|---|---|
| MICTOR.31 | AK11 | |
| MICTOR.32 | AK14 | PPC_TRC_TS3 |
| MICTOR.33 | AJ11 | |
| MICTOR.34 | AL14 | PPC_TRC_TS4 |
| MICTOR.35 | AM11 | |
| MICTOR.36 | AK13 | PPC_TRC_TS5 |
| MICTOR.37 | AL11 | |
| MICTOR.38 | AL13 | PPC_TRC_TS6 |

## 8.8 RS232

One RS232 port (P1) is provided for use by the user design. This RS232 port can be accessed from any FPGA, but only by one FPGA at a time. This multiplexed port can be configured from the USB GUI Windows application.

| Signal Name | Pin on FPGA A | Pin on FPGA B | Pin on FPGA C |
|---|---|---|---|
| RS232_RX | AF1 | AK16 | AH17 |
| RS232_TX | AE1 | AJ16 | AJ17 |

# 9 LEDs

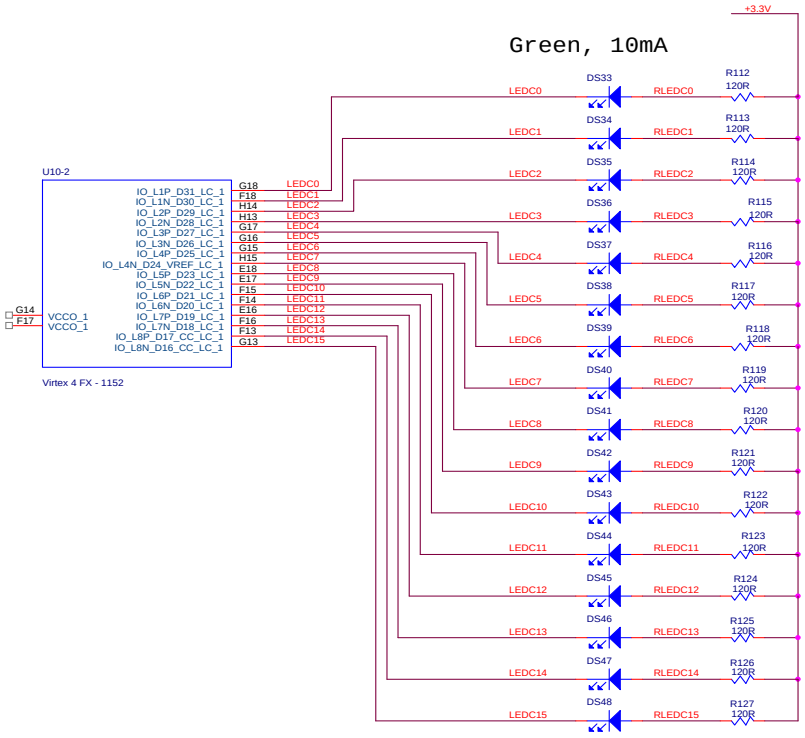



Figure 32 FPGA C LEDs

FPGA A is connected to 8 green LEDs. FPGA C is connected to 16 LEDs. These LEDs can be used for the user design. The brightness of these LEDs can be controlled by changing the output standard on the LED signals from 2, 4, 12, 16 or 24mA.

| Signal Name | Pin on FPGA A | Pin on FPGA B | Pin on FPGA C |
|---|---|---|---|
| LED0 | AJ6 | | G18 |
| LED1 | AJ5 | | F18 |
| LED2 | AG10 | | H14 |
| LED3 | AK6 | | H13 |
| LED4 | AH10 | | G17 |
| LED5 | AH9 | | G16 |
| LED6 | AK7 | | G15 |
| LED7 | AJ7 | | H15 |
| | | | |
| LED8 | | | E18 |
| LED9 | | | E17 |
| LED10 | | | F15 |
| LED11 | | | F14 |
| LED12 | | | E16 |

| | | | |
|---|---|---|---|
| LED13 | | F16 | |
| LED14 | | F13 | |
| LED15 | | G13 | |

# 10 RocketIO

## 10.1 RocketIO Clock Resources

Since it is impossible to determine during manufacturing the clocking requirements of every possible end application, the DN8000K10PCI comes with a flexible clock network capable of a wide range of serial frequencies, while maintaining the tight jitter requirements of the 10 Gigabit serial transceivers.

The RocketIO clock tree is driven by a synthesizer and two oscillators, and dedicated multiplexers inside the Virtex 4 FPGA allow the user to switch between these clock sources.

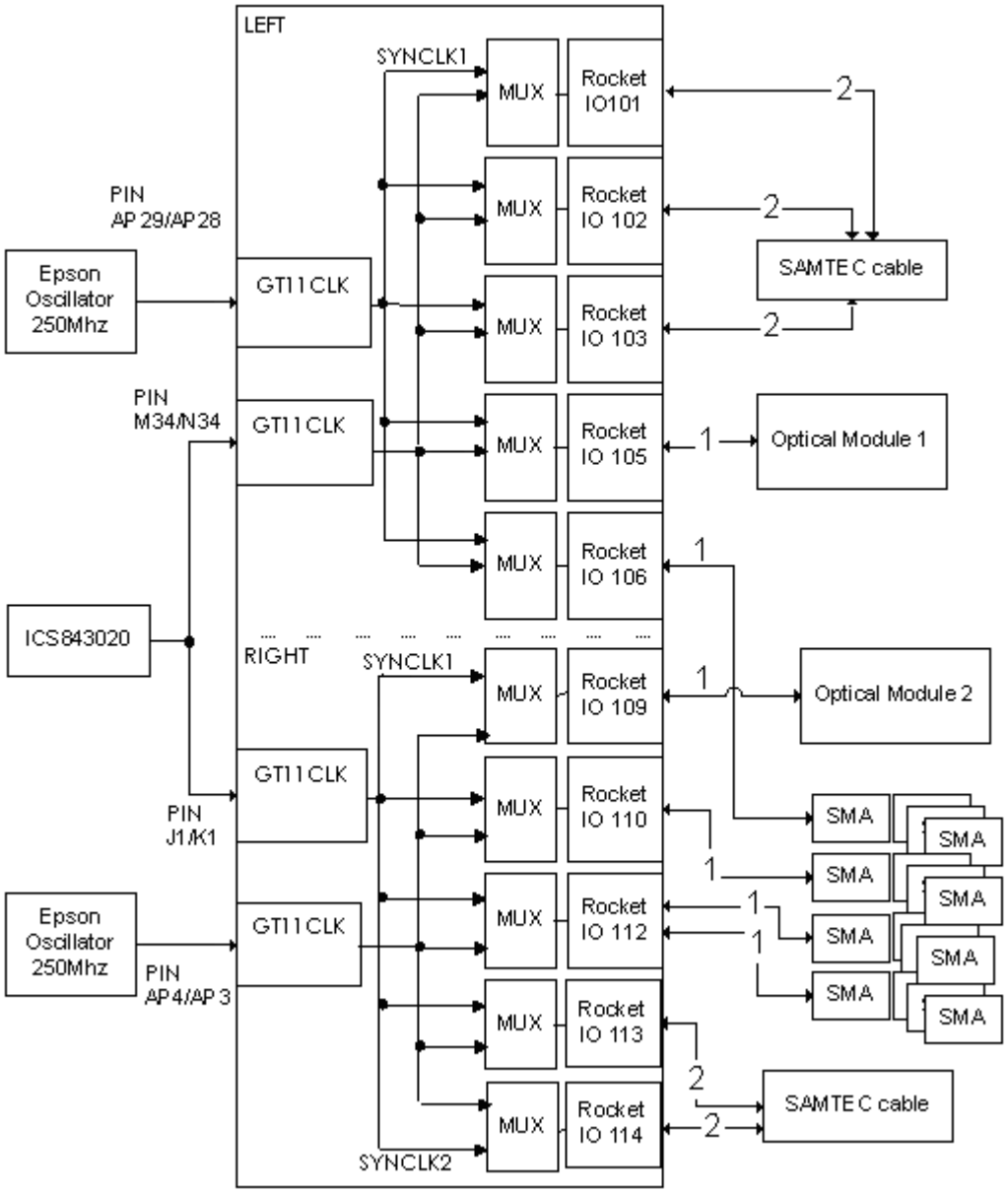



Figure 33 Internal MGT clocking

The RocketIOs on the Virtex 4 FPGA is divided into two columns, X0 and X1. The clock network of each column is separate and clocks may not be shared between the two columns. Each column has two clock distribution trees and two clock inputs. Each tree can be driven by a clock input, by a clock from a global clock input
(not recommended) or by a recovered clock. Finally, each tile has a multiplexer than can select from one of the two clock trees to clock that entire tile (REFCLK1 or REFCLK2). Use the TXABPMACLKSEL and RXPMACLKSEL parameters.

The diagram above shows the two RocketIO columns and the connectivity of each.

Once a clock is routed to an MGT tile, that clock can be multiplied and divided by the MGT tile.

Most users will want to use the frequency synthesizer for generating RocketIO reference clocks. The ICS843020-01 synthesizer is very low jitter and should suitable for operation up to 6Gbs RocketIO operation. The frequency of the synthesizer can be adjusted through the main.txt file on the SmartMedia card, or through the USB GUI program.

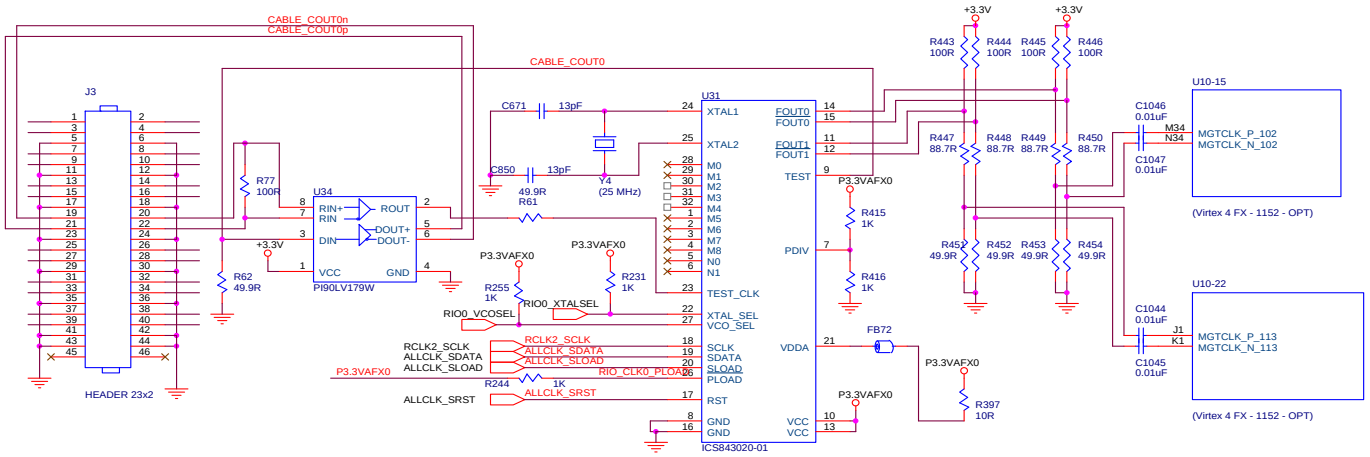



Figure 34 MGT 8442 Connections

The LVPECL outputs of the ICS843020 are scaled down to meet the input requirements of the MGTCLK inputs.

An output from the ICS843020-01 is also converted to LVDS and driven to J3 pins 19 and 21, the Samtec QSE-DP connector. This can be used to forward a RocketIO clock off board along with rocketIO signals to support standards that require an exact reference clock, like PCI Express. J3 may also drive pins 20 and 22. The ICS843020-01 can receive this clock and use it to generate a frequency for the MGTCLK inputs.

For 10Gb serial transmission rates, you should use one of the low-jitter fundamental frequency SAW oscillators. These oscillators operate at 250Mhz and so cover the gaps in the frequency synthesis options given by the ICS843020-01.

**Error!**

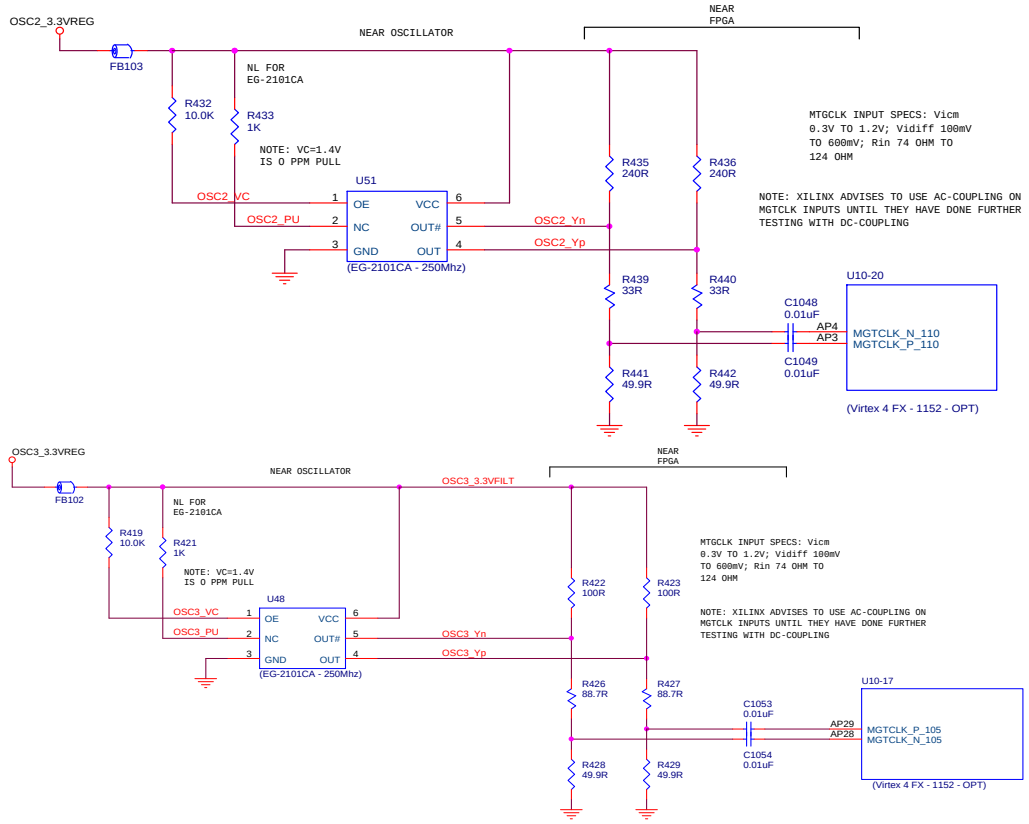



Figure 35 MGT PECL Oscillators

There are two Epson2101CA SAW oscillators, U51 and U48. Each one drives a MGTCLK on to one side of the

The ICS843020-01 Frequency Synthesizer is a very low phase noise. With the default 25Mhz oscillator, the frequency synthesizer is capable of producing frequencies in the ranges 71.875-84.375, 143.75-168.75, 287.5-337.5, and 575-675 Mhz.

## 10.2 MGT Power network

The RocketIO strict power supply constraints require the use of heavy power supply filtering. The RocketIO's three power rails are each generated by a linear voltage regulator.

### 10.2.1 FX CES2 rework

If your DN8000K10PCI came with the option "FPGA C – FX60CES2", then a late-breaking Virtex 4 erratum required the following rework. This rework is not shown in the schematic on the CD.

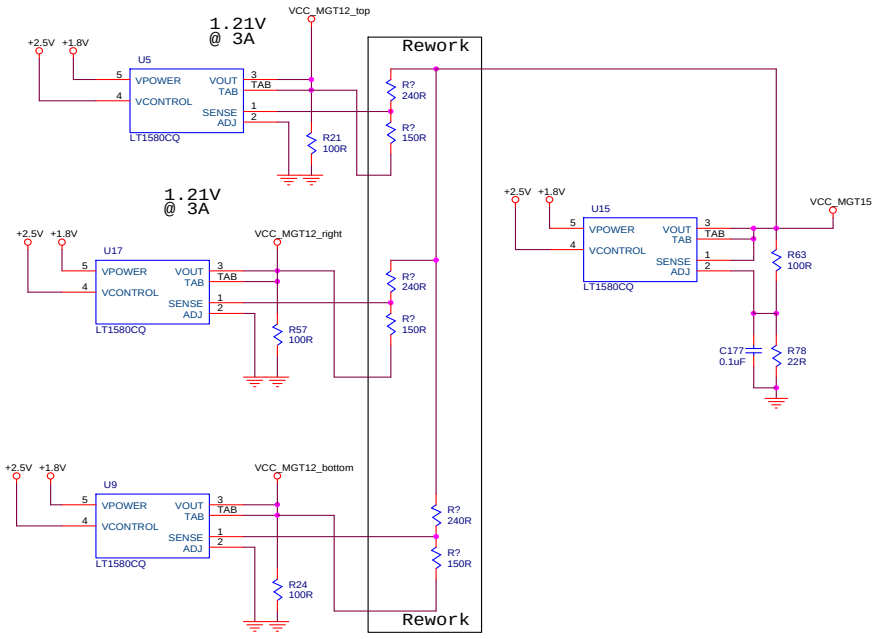



Figure 36 MGT 1.1V rework

This rework drops the 1.2V RocketIO supply from 1.25V to 1.14V.

This rework only appears on revision 2 boards that have a Virtex-4 FX60 CES2 or CES3 in the FPGA C position.

## 10.3 The connections

The following is a connection summary of the RocketIO channels available on the DN8000K10PCI

| Group Name | | Signal Name | Pin on FPGA C | Tile Number | MGT tile FPGA C |
|---|---|---|---|---|---|
| QSE cable 0 | J2 | QSE01_TXP | C1 | 113 | GT11_X1Y6 |
| | | QSE01_TXN | D1 | | |
| | | QSE01_RXP | F1 | | |
| | | QSE01_RXN | G1 | | |
| | | QSE02_TXP | A4 | 113 | GT11_X1Y7 |
| | | QSE02_TXN | A3 | | |
| | | QSE02_RXP | A7 | | |
| | | QSE02_RXN | A6 | | |
| | | QSE03_TXP | A13 | 114 | GT11_X1Y8 |
| | | QSE03_TXN | A12 | | (FX100) |
| | | QSE03_RXP | A10 | | NOT ON FX60 |
| | | QSE03_RXN | A9 | | |
| | | QSE04_TXP | A15 | 114 | GT11_X1Y9 |

| | | | | |
|---|---|---|---|---|
| | QSE04_TXN | A14 | | (FX100) |
| | QSE04_RXP | A18 | | |
| | QSE04_RXN | A17 | | NOT ON FX60 |
| | | | | |
| | QSE11_TXP | A25 | 101 | GT11_X0Y8 |
| QSE J3 | QSE11_TXN | A26 | | (FX100) |
| cable 0 | QSE11_RXP | A28 | | |
| J3 | QSE11_RXN | A29 | | NOT ON FX60 |
| | QSE12_TXP | A23 | 101 | GT11_X0Y9 |
| | QSE12_TXN | A24 | | (FX100) |
| | QSE12_RXP | A20 | | |
| | QSE12_RXN | A21 | | NOT ON FX60 |
| | QSE13_TXP | F34 | 102 | GT11_X0Y6 |
| | QSE13_TXN | G34 | | |
| | QSE13_RXP | J34 | | |
| | QSE13_RXN | K34 | | |
| | QSE14_TXP | D34 | 102 | GT11_X0Y7 |
| | QSE14_TXN | E34 | | |
| | QSE14_RXP | A31 | | |
| | QSE14_RXN | A32 | | |
| | QSE15_TXP | Y34 | 103 | GT11_X0Y4 |
| | QSE15_TXN | AA34 | | |
| | QSE15_RXP | AC34 | | |
| | QSE15_RXN | AD34 | | |
| | QSE16_TXP | V34 | 103 | GT11_X0Y5 |
| | QSE16_TXN | W34 | | |
| | QSE16_RXP | R34 | | |
| | QSE16_RXN | T34 | | |
| | | | | |
| | XFP1_TXDP | AL34 | 105 | GT11_X0Y2 |
| | XFP1_TXDN | AM34 | | |
| | XFP1_RXDP | AP32 | | |
| | XFP1_RXDN | AP31 | | |
| | | | | |
| | XFP1_REFCLK | AJ34 | 105 | GT11_X0Y3 |
| | XFP1_REFCLKN | AK34 | | |
| | | | | |
| | XFP1_INTERRUPT_N | AD9 | | |
| | XFP1_MOD_ABS | AK9 | | |
| | XFP1_MOD_DESEL | AE14 | | |
| | XFP1_MOD_NR | AL9 | | |
| | XFP1_P_DOWN | AD12 | | |
| | XFP1_RX_LOS | AE12 | | |
| | XFP1_SCL | AE11 | | |
| | XFP1_SDA | AD11 | | |

| | | | | |
|---|---|---|---|---|
| | XFP1_TX_DIS | AD10 | | |
| | XFP2_TXP | AP9 | 109 | GT11_X1Y1 |
| | XFP2_TXN | AP10 | | |
| | XFP2_RXP | AP6 | | |
| | XFP2_RXN | AP7 | | |
| | XFP2_REFCLK | AP11 | 109 | GT11_X1Y0 |
| | XFP2_REFCLKN | AP12 | | |
| | XFP2_INTERRUPT_N | AG15 | | |
| | XFP2_MOD_ABS | AK16 | | |
| | XFP2_MOD_DESEL | AF15 | | |
| | XFP2_MOD_NR | AF10 | | |
| | XFP2_P_DOWN | AH15 | | |
| | XFP2_RX_LOS | AG10 | | |
| | XFP2_SCL | AJ10 | | |
| | XFP2_SDA | AH10 | | |
| | XFP2_TX_DIS | AJ16 | | |
| | SFP1_TXDP | AJ34 | 105 | GT11_X0Y3 |
| | SFP1_TXDN | AK34 | | |
| | SFP1_RXDP | AF34 | | |
| | SFP1_RXDN | AG34 | | |
| | SFP1_TXDIS | AA11 | | |
| | SFP1_LOS | AH8 | | |
| | SFP1_RATE_SEL | AA13 | | |
| | SFP1_TXFAULT | AB11 | | |
| | SFP1_MOD-DEF0 | AB13 | | |
| | SFP1_MOD-DEF1 | AJ7 | | |
| | SFP1_MOD-DEF2 | AK7 | | |
| | SFP2_TXDP | AP9 | 109 | GT11_X1Y0 |
| | SFP2_TXDN | AP10 | | |
| | SFP2_RXDP | AP6 | | |
| | SFP2_RXDN | AP7 | | |
| | SFP2_TXDIS | AM8 | | |
| | SFP2_LOS | AK8 | | |
| | SFP2_TXFAULT | AB12 | | |
| | SFP2_RATE_SEL | AL8 | | |
| | SFP2_MOD-DEF0 | AC13 | | |
| | SFP2_MOD-DEF1 | AD14 | | |

| | | | | |
|---|---|---|---|---|
| | SFP2_MOD-DEF2 | AM7 | | |
| SMA | | | | |
| connector J22 | RIO_SMA_TXP0 | AP21 | 106 | GT11_X0Y0 |
| J23 | RIO_SMA_TXN0 | AP20 | | |
| J20 | RIO_SMA_RXP0 | AP18 | | |
| J21 | RIO_SMA_RXN0 | AP17 | | |
| | | | | |
| J16 | RIO_SMA_TXP1 | AH1 | 110 | GT11_X1Y2 |
| J17 | RIO_SMA_TXN1 | AJ1 | | |
| J18 | RIO_SMA_RXP1 | AL1 | | |
| J19 | RIO_SMA_RXN1 | AM1 | | |
| | | | | |
| J33 | RIO_SMA_TXP2 | R1 | 112 | GT11_X1Y5 |
| J34 | RIO_SMA_TXN2 | T1 | | |
| J31 | RIO_SMA_RXP2 | M1 | | |
| J32 | RIO_SMA_RXN2 | N1 | | |
| | | | | |
| J25 | RIO_SMA_TXP3 | U1 | 112 | GT11_X1Y4 |
| J26 | RIO_SMA_TXN3 | V1 | | |
| J27 | RIO_SMA_RXP3 | Y1 | | |
| J28 | RIO_SMA_RXN3 | AA1 | | |
| Other | | | | |
| | Loopback_TXp | AP23 | 106 | X0Y1 |
| | Loopback_TXn | AP22 | | |
| | Loopback_RXp | AP26 | | |
| | Loopback_RXn | AP25 | | |
| | | | | |
| | NC_TXp | AF1 | 110 | GT11_X1Y3 |
| | NC_TXn | AG1 | | |
| | NC_RXp | AC1 | | |
| | NC_RXn | AD1 | | |

## 10.4 Samtec Multi Gigabit ribbon cable

For board-to-board high-density connections, two Samtec ribbon cable connectors (J2 and J3) are connected to RocketIO. The pinouts on the cable allow two DN8000K10PCI boards to be connected to each other for a total of 10 bi-directional channels operating at 5Gbs per channel, per direction.

The Samtec part number (J2, J3) QSE-014-01-F-D-DP-A

An appropriate crossover cable for cabling two DN8000K10PCIs together is the Samtec EQDP-014-09.00-TBR-TBL-4

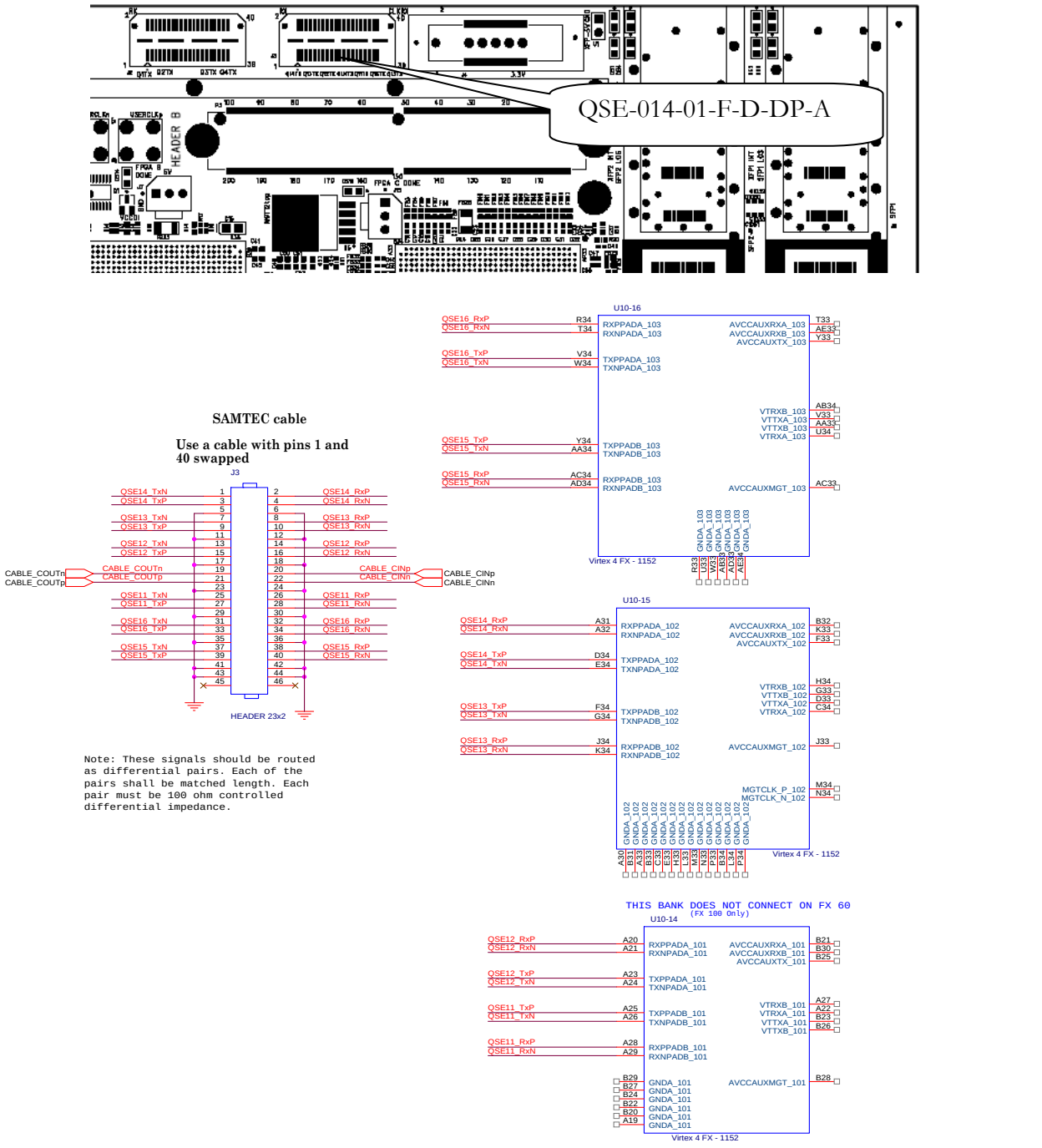
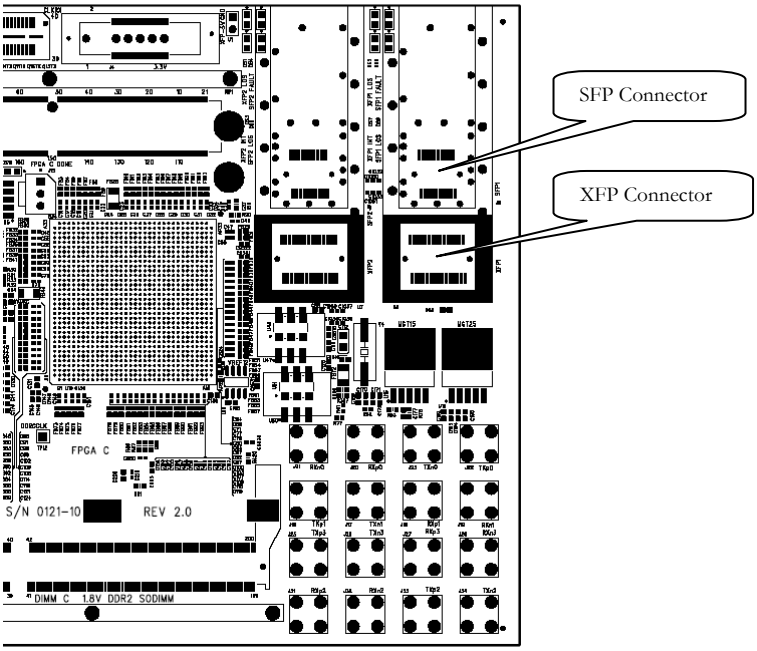

QSE-014-01-F-D-DP-A
SAMTEC cable
Use a cable with pins 1 and 40 swapped
J3
HEADER 23x2
Note: These signals should be routed as differential pairs. Each of the pairs shall be matched length. Each pair must be 100 ohm controlled differential impedance.
U10-16
U10-15
U10-14
Virtex 4 FX - 1152
THIS BANK DOES NOT CONNECT ON FX 60
(FX 100 Only)

Figure 37 QSE Connector

Each connector also has a clock input that can be routed to the MGT CLK of FPGA C to allow standards that require transmitting at an exact frequency, such as PCI Express.

## 10.5 Optical Modules

The DN8000K10PCI comes with two optical module connectors. If you need to interface to a specific standard, the easiest way is to buy an SFP or XFP module that supports that standard.




### 10.5.1 SFP

SFP modules support 1-4.5Gbs serial transmission rate. There are two module options available on the DN8000K10PCI, none, one or both of which can be SFP (or XFP). This is a manufacturing-time ordering option

Two red LEDs show the status of the channel. The LOS LED indicates that the far end transmitter is not operating, the cables are not secured or matched to the transmitter wavelength. The INT LED indicates. The FAULT LED indicates a transmission laser failure, or an unsecured module.

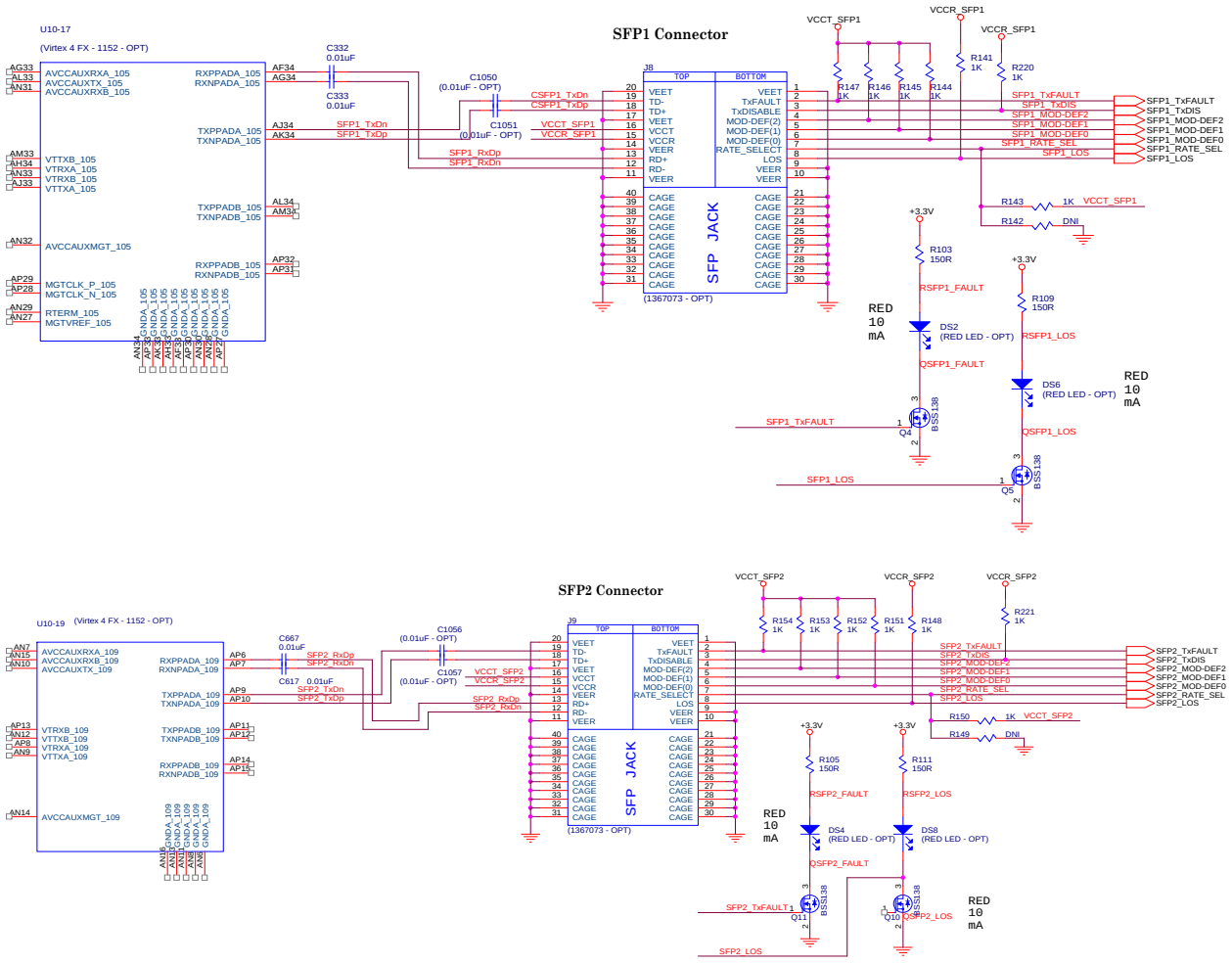
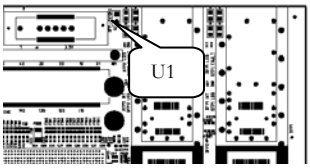


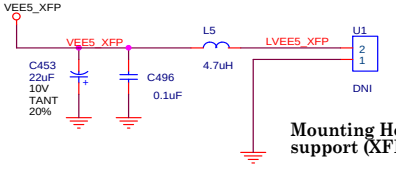

Figure 38 SFP modules

### 10.5.2 XFP

XFP modules are the only 10Gbs optical module that allows for multiple signaling standards. There are two module options available on the DN8000K10PCI, none, one or both of which can be XFP (or SFP). This is a manufacturing-time ordering option.

The XFP specification allows for an optional –5.2V power supply to be provided by the host board for ECL transmitter modules. The DN8000K10PCI provides no –5.2V power, so a mounting point (U1) is provided for the use of a bench supply if ECL signaling is required.



Some XFP modules may require a reference clock to retime the transmitted signal (The REFCLK signal in the XFP specification). The REFCLK signal is connected to a RocketIO output on FPGA C. The REFCLK signal should be 1/64 of the data rate driven onto the XFP's TX pins. To drive this signal, See Xilinx Application note XAPP656. To meet the input requirements of the XFP module, you must increase the differential swing voltage of the MGT transmitter outputs. Set TXDAT_TAP_DAC to 800mV.

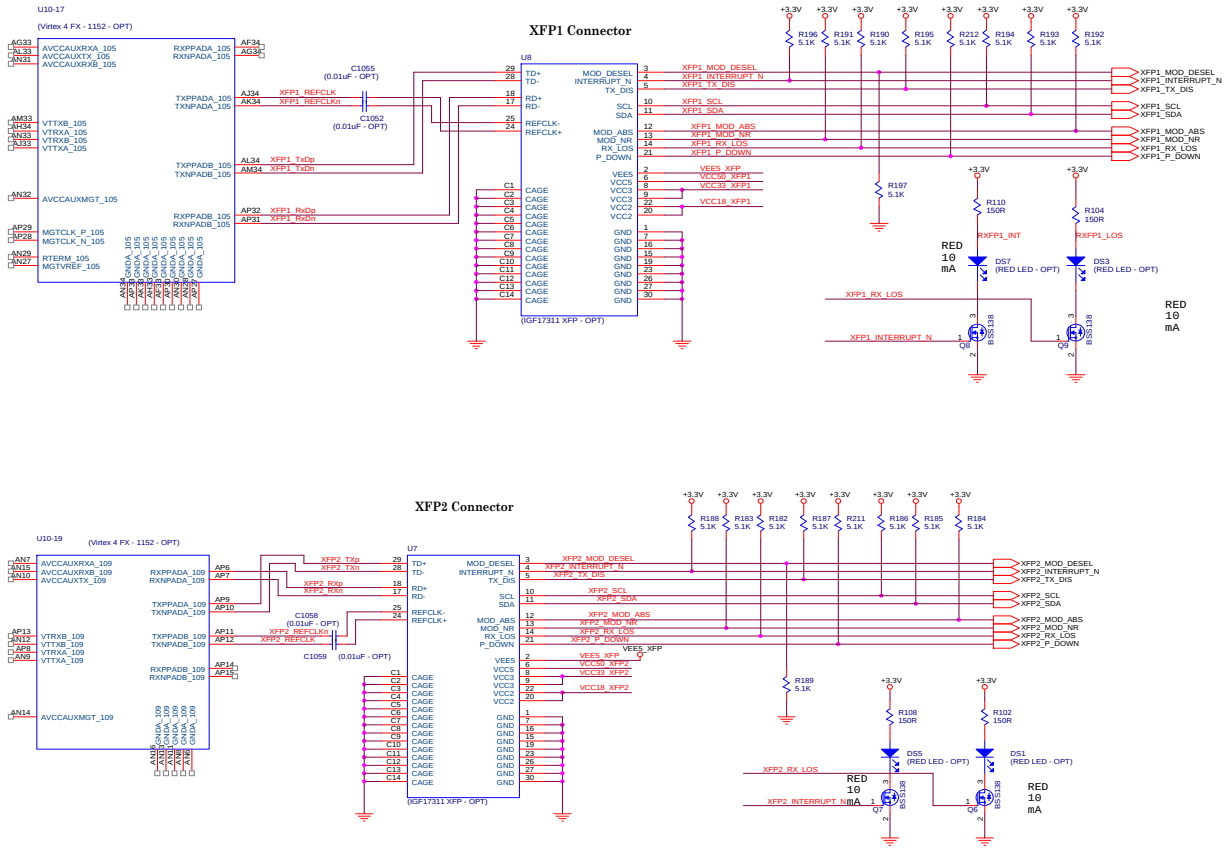



Figure 39 XFP Modules

The XFP interface is tested with the module Intel TXN18107 at 3.25Gbs. This module can be purchased from insight.com. At the time of printing, the Virtex-4 MGTs are only capable of 3.25Gbs transmission. Note that 3.25 Gbps is not supported by the XFI specification, so Dini Group cannot guarantee the XFP interface will interoperate with other equipment.

The XFP refclock signal is driven from

## 10.6 The SMAs

The easiest way to connect two RocketIO channels is through the use of SMA cables. The SMA connections on the DN8000K10PCI were designed to operate at the full 11Gb potential of the Virtex 4 RocketIO transceivers.

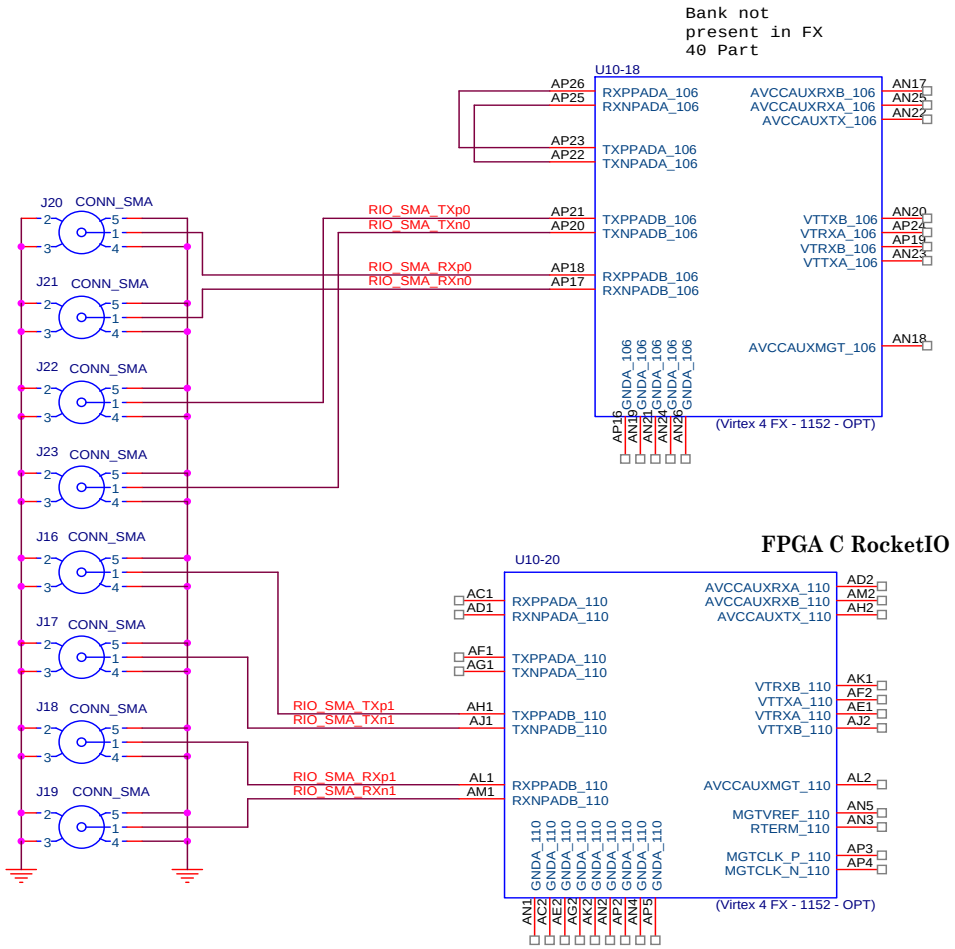


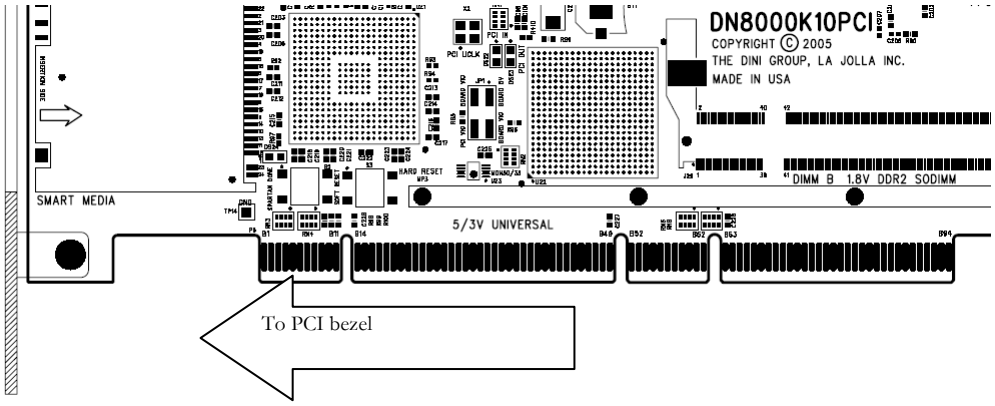

Figure 40 SMA Connections

The loopback pair AP26 and AP25 can be used to test your Virtex 4 fabric design. You may want to get the loopback pair working before attempting to transmit high data rates over a cable system.

# 11 PCI interface

## 11.1 PCI edge connector

3.3v or 5.0V universal 64-bit card edge connector.




Do not attempt to plug a PCI card in backwards if it does not fit.

## 11.2 The Quicklogic 5064

In order to provide a high speed easy-to-use interface for your design, the DN8000K10PCI comes equipped with a PCI bridge, a Quicklogic 5064. The interface to the Quicklogic 5064 is a simple FIFO. In order to use this interface, you should implement the PCI interface module supplied with the reference design. (Blockram Access A)

A description of this module is maintained in a separate document on the user CD, QL5064_Interface_Module.pdf
D:\FPGA_Reference_Designs\dn8000k10pci\PCI_interface\QL5064_Interface_Module.pdf

The rest of this section describes to function of the PCI hardware.

## 11.3 Virtex 4 FPGA Communication

When the board is in a PCI slot, whether or not FPGA A is configured, the device will communicate with the host. This is because the QL5064 is always active. The QL5064 will report a Vendor ID: 17DF, and a Device ID: 1864.

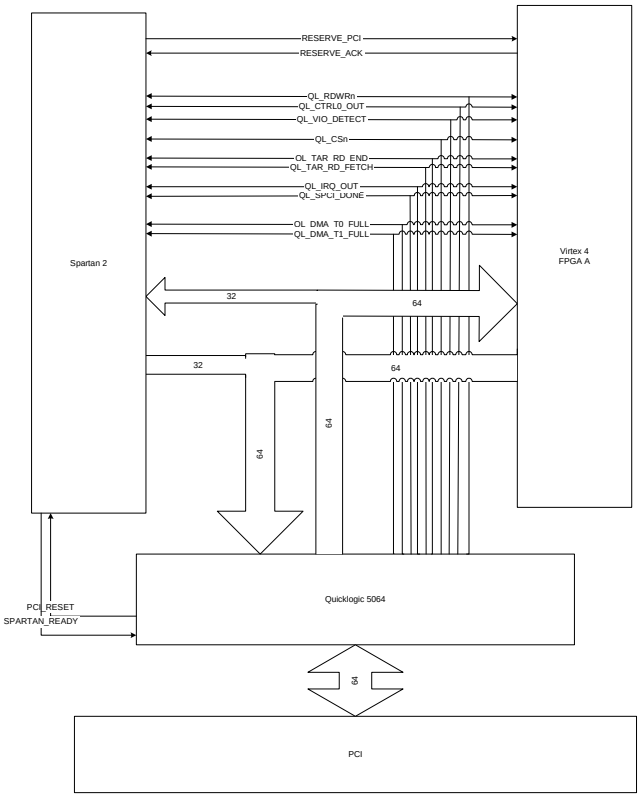
The QL5064 will define BARs 0 through 7.

BARs 1-5 are reserved for user communication with the FPGA.

The size of the memory space for the BARs is as follows:
BAR 0: 8MB
BAR 1-5: 16MB

## 11.4 Spartan 2 Communication

The QL5064 will also define BAR 0. This range is reserved for the Dini Group reference design, configuration, and testing purposes. The only reason you should use BAR 0 is to communicate over the Main Bus interface.




The above diagram shows the signal connections between the Quicklogic, Cyclone and Stratic FPGA. Data from BAR 1-5 is transferred from the QL5064 to the Stratix FPGA over the 64-bit QLDATA_IN and QLDATA_OUT busses.

Data on BAR 0 is transferred from the QL5064 to the Cyclone FPGA over the lower 32 bits (QLDATA_IN0-31, QLDATA_OUT0-31).

To access the "MainBus" interface, follow these steps:

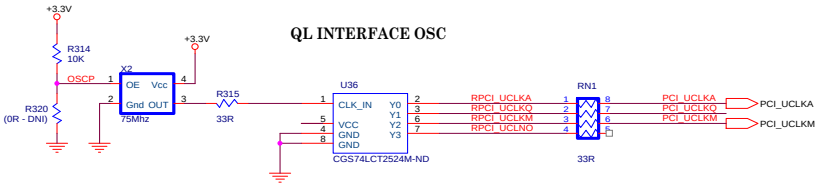
6) Write the 32-bit (4 byte) "MainBus" address you wish to access to BAR 0, offset 0x240. This sets the Cyclone's main bus address register. This also causes a "main bus address latch" to occur over main bus.

7) To read from main bus at this address read 4 bytes from BAR 0, offset 0x250. This will cause the Cyclone to read from main bus and return the 32-bit result. After this is done, the Cyclone will increment the address in its address register, so consecutive addresses can be read by reading from this offset multiple times. It will then cause a "main bus address latch" to occur.

To write to an address on main bus, write 4 bytes to BAR 0, offset, 0x248. After a write is completed, the Address register incrememnts by 1., then, a "main bus address latch" will occur.

## 11.5 PCI clocking

All communication to the Quicklogic 5064 chip is synchronized with a 75Mhz oscillator. The 75Mhz PCI UCLK is delivered to FPGA A, the Spartan 2 FPGA and the Quicklogic 5064.

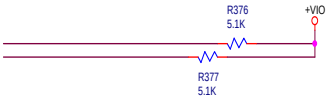



-

## 11.6 JTAG

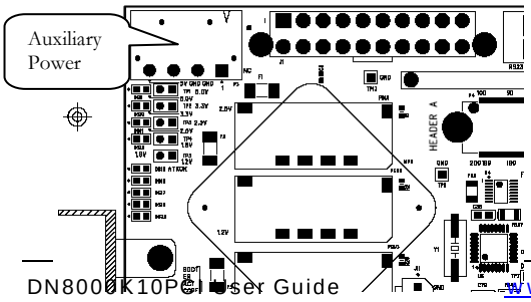
The PCI connector's JTAG signals are looped back to bypass the DN8000K10PCI when it is plugged into the PCI slot.

SDONE, SBO-




## 11.7 PCI Power

In most applications, the DN8000K10PCI can draw its power from the PCI slot. The PCI specification requires that the motherboard provide 25W of 5V power for the DN8000K10PCI to use (Most motherboards provide well in excess of this amount, supplying the power for PCI cards directly from the ATX power supply). In high power applications exceeding 25W, you may need to connect the Auxiliary power connector (P3).

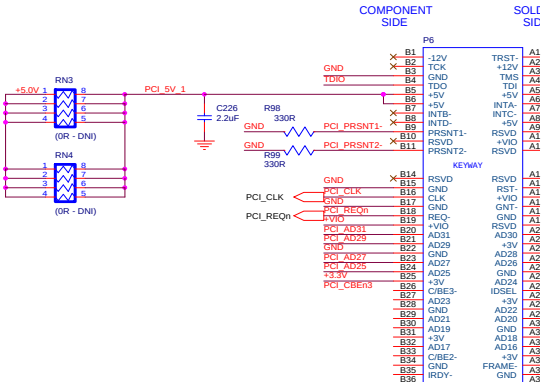



The Aux. Power connector is a standard IDE hard drive power connector and should be supplied by the ATX power supply that is in your computer case. Aux power connector 5.0V and 3.3V are

shorted to the PCI slot 5.0V and 3.3V. **The power supply driving the PCI slot and IDE power cable must be driven from the same unit.**

If you are operating the DN8000K10PCI in a server or other enclosure that does not have available IDE hard drive cables, and you intend to use the DN8000K10PCI in a high-power application, then an alternate setup will be required. There are eight 0Ω resistors shorting the PCI edge connector's 5V pins and the power distribution plane on the DN8000K10PCI. This connection disallows safely connecting the Aux. IDE power connector to an external power supply. To allow the use of a separate power supply dedicated to the DN8000K10PCI while it is in a PCI slot, these resistors can be removed. (*Please e-mail tech support if you think that you need these resistors removed*). You will also need remove the 12V wire on the IDE hard drive cable to prevent contention with the PCI slot's 12V supply. Note that in this configuration, the 5.0V power for the DN8000K10PCI (and most of the required power) will be supplied from the external supply over P3. The 3.3V power for the DN8000K10PCI will continue to be supplied from

The eight 0Ω resistors are RN3, RN4, RN6, RN5, RN47, RN48, RN50, RN51

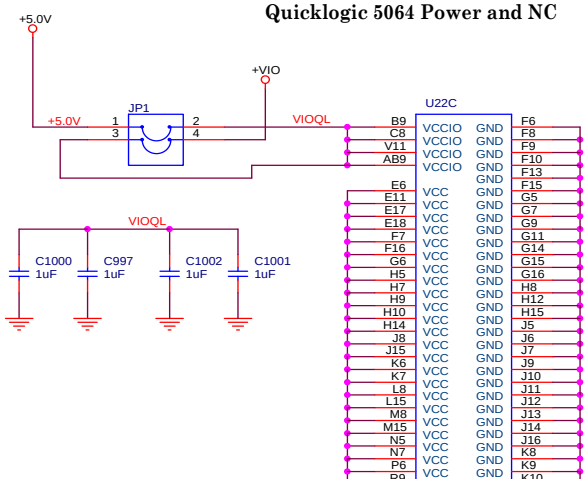



Remember, Most users will use the same power supply to power the motherboard and Aux. IDE power connector (P3). In this configuration, these modifications are not required.

## 11.8 PCI Signaling

To allow universal (3.3V or 5.0V) PCI IO, the DN8000K10PCI uses the PCI bus's VIO pins to detect the IO levels used by your motherboard. Most motherboards use 5V signal levels on the

PCI bus, but many servers, and all 64bit PCI slots require 3.3V signaling levels.



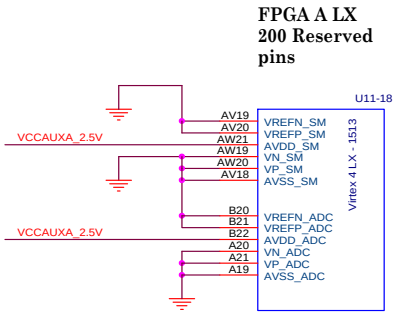

IO voltage for the DN8000K10PCI is provided by a jumper connecting the PCI slot's VCCIO signal to the Quicklogic 5064. For correct operation in stand-alone mode, a jumper should be installed between JP1.1 and JP1.3. For correct operation in PCI mode, a jumper should be installed between JP1.2 and JP1.4

You should never install a jumper in both positions. This could short 5.0V to 3.3V when plugged in to a 64-bit, 3.3V slot.

The DN8000K10PCI can be used in a PCI or PCI-X slot operating at 33Mhz or 66Mhz. It can also be used in 100Mhz and 133Mhz busses, although the DN8000K10PCI will cause the entire bus to operate at 66Mhz.

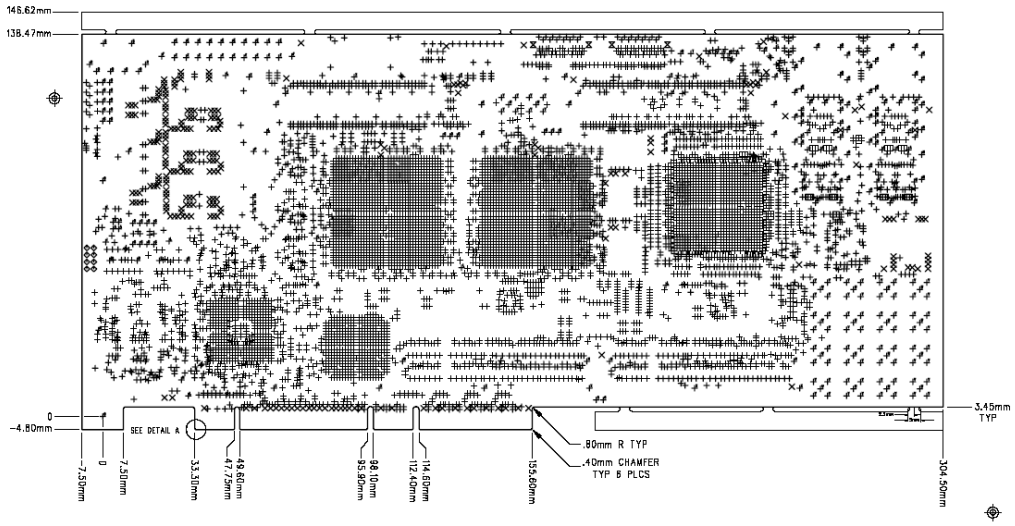
# 12 FPGA System monitor/ADC

The System Monitor and ADC functions of the Virtex 4 FPGA are no longer supported by Xilinx. The most important responsibility of the System Monitor, temperature sensing, has been moved to the configuration circuitry. The DN8000K10PCI will automatically monitor and prevent thermal overload in the three Virtex 4 FPGAs. No user action is required.



# 13 Mechanical

The dimensions of the PWB are 312mm long by 135mm tall, plus a 8.25mm PCI edge connector. This is taller than the PCI specification allows, although the DN8000K10PCI fits easily inside most ATX computer cases.




The topside clearance with the factory installed active heatsinks is 23mm. This leaves just enough room for airflow if the adjacent PCI slot is left unoccupied, or the DN8000K10PCI is the last PCI card in the row. The default heatsinks can be removed if you do not require high-power operation, allowing the DN8000K10PCI to meet the PCI height restriction. The back-side clearance is 3.5mm. This exceeds the PCI specification by 2.5mm.

If it is required that the DN8000K10PCI use only one PCI slot, the fan can be removed from the active heatsink assembly, as long as sufficient airflow is provided. Most PC cases do not provide sufficient airflow for high-power applications.

The board should plug into any PCI or PCI-X slot with 5V or 3.3V keying, 32-bit or 64-bit slot widths. (33Mhz or 66Mhz (100 and 133Mhz will be brought down to 66Mhz automatically

Chapter 5

# Introduction to the Reference Design

*This chapter introduces the DN8000K10PCI Reference Design, including information on what the reference design does, how to build it from the source files, and how to modify it for another application.*

## 1 Exploring the Reference Design

### 1.1 What is the Reference Design?

The reference design is a fully functional Virtex 4 FPGA design capable of demonstrating most of the features available on the DN8000K10PCI. Features exercised in the reference design include:

- Access to the DDR2 SDRAM Modules At 200Mhz
- UART Communication
- FPGA Interconnect
- Interaction with the Configuration FPGA and MCU
- Use of Embedded PowerPC Processors (eventually)
- Memory Mapped Access Between PPC And User Design (eventually)
- Access to external LEDs
- Communication via Rocket I/O Transceivers
- Instantiation of Daughter Card Test Headers
- USB memory map to DDR2 memory.
- PCI memory map to DDR2 memory.

- Pin-multiplexed FPGA interconnect using LVDS at 700Mbs per signal pair

All source code for the reference design is included on the CD and may be used freely in customer development. Precompiled bit files for the most common stuffing options are also included and can be used to verify board functionality before beginning development. A build utility, described in the section Compiling The Reference Design, can be used to generate new bit files, or to generate bit files for less common configurations of the DN8000K10PCI.

The reference design was created using

Here are the default main.txt file lines.

```
verbose level: 2
sanity check: y
clock frequency: A N 4 M 16 // 100 MHz – not used for PCI/Main test
clock frequency: B N 2 M 28 // 200 MHz
clock frequency: D N 2 M 25 //  200 MHz
clock frequency: 2 N 2 M 25 // 312 MHz
```

# 2 Reference Design Memory Map

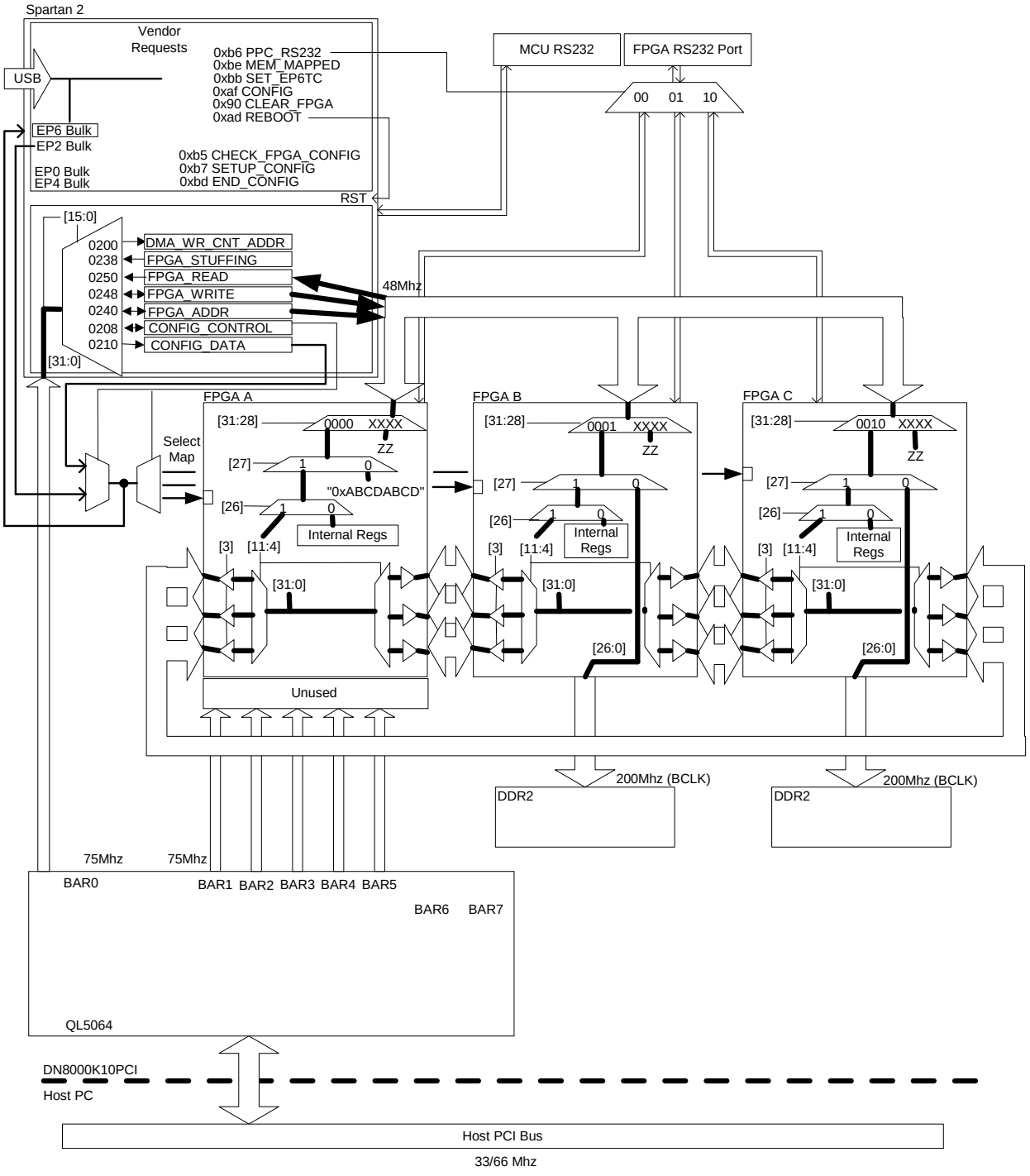
The Dini Group reference design memory maps the main features of the DN8000K10PCI to the host interfaces: PCI, USB, and RS232. This memory map applies to the "Main" reference design, and may have differences for the "RocketIO", "LVDS" and "PCI" reference design.

The Main Bus interface is used to access the reference design memory map. Addresses are 32-bits. Each address contains a 32-bit word.

| | | | |
|---|---|---|---|
| FPGA A | 0x00000000 - 0x0FFFFFF | | Not defined |
| FPGA A | 0x08000002 | IDCODE | 0x05000121 |
| FPGA A | 0x08000004 | INTERCONTYPE | 0x34561111 |
| FPGA A | 0x08000006 | RWREG | Scratch Register for testing |
| FPGA A | 0x08000010 | LED_OE | Controls LED output enables |
| FPGA A | 0x08000011 | LED_OUT | Controls LED outputs |
| FPGA A | 0x08100001 | CLK_COUNTER | Contains contents of ACLK counter |
| FPGA A | 0x08100002 | CLK_COUNTER | Contains contents of BCLK counter |
| FPGA A | 0x08100003 | CLK_COUNTER | Contains contents of DCLK counter |
| FPGA A | 0x08100004 | CLK_COUNTER | Contains contents of SYSCLK counte |

| | | | |
|---|---|---|---|
| FPGA A | 0x0C000000 | ABP0 OUT | W; the output state of FPGA IOs connected to the ABP0 interconenct bus |
| FPGA A | 0x0C000004 | ABP0 OE | W; The ouput enable of each FPGA IO on the ABP0 interconnect bus. |
| FPGA A | 0x0C000008 | ABP0 IN | The input state of each FPGA IO… …on the ABP0 interconnect bus |
| FPGA A | 0x0C00000C | ABP0 Name | "ABP0" (ascii) |
| FPGA A | 0x0C000010 | ABP1 OUT | W; ABP1 IO output values |
| FPGA A | 0x0C000014 | ABP1 OE | W; Output enable of ABP1 bus |
| FPGA A | 0x0C000018 | ABP1 IN | R; ABP1 input values |
| FPGA A | 0x0C00001C | ABP1 Name | "ABP1" (acsii) |
| | | | |
| FPGA A | 0x0C000XX0 | BUS XX OUT | XX can be 0-21 hex. Output status of IOs on bus XX. |
| FPGA A | 0x0C000XX4 | BUS XX OE | XX can be 0-21 hex. OE status of IOs |
| FPGA A | 0x0C000XX8 | BUS XX IN | XX can be 0-21 hex. The input values |
| FPGA A | 0x0C000XXC | BUS XX Name | The name of the bus XX (schematic) |
| | | | |
| FPGA B | 0x10000000 - | DDR2 B space… | Mapped to DDR2 SODIMM… |
| FPGA B | 0x17FFFFFF | … | …interface |
| | | | |
| FPGA B | 0x18000002 | IDCODE | 0x05000121 |
| FPGA B | 0x18000004 | INTERCONTYPE | 0x34561111 |
| FPGA B | 0x18000006 | RWREG | Scratch Register for testing |
| FPGA B | 0x18000010 | LED_OE | Controls LED output enables |
| FPGA B | 0x18000011 | LED_OUT | Controls LED outputs |
| FPGA B | 0x18100001 | CLK_COUNTER | Contains contents of ACLK counter |
| FPGA B | 0x18100002 | CLK_COUNTER | Contains contents of BCLK counter |
| FPGA B | 0x18100003 | CLK_COUNTER | Contains contents of DCLK counter |
| FPGA B | 0x18100004 | CLK_COUNTER | Contains contents of SYSCLK counte |
| | | | |
| FPGA B | 0x18000001 | DDR2HIADDR | upper address bits for DDR2 interface |
| FPGA B | 0x18000003 | HIADDRSIZE | number of bits in DDR2HIADDR |
| FPGA B | 0x18000005 | DDR2SIZEHIADDR | The size of the DDR2 module. |
| FPGA B | 0x18000007 | DDR2TAPCNT0 | Current IDELAY values of DDR2… |
| FPGA B | 0x18000008 | DDR2TAPCNT1 | …interface |
| | | | |
| FPGA B | 0x1C000XX0 | BUS XX OUT | XX can be 0-21 hex. Output status of IOs on bus XX. |
| FPGA B | 0x1C000XX4 | BUS XX OE | XX can be 0-21 hex. OE status of IOs |
| FPGA B | 0x1C000XX8 | BUS XX IN | XX can be 0-21 hex. The input values |
| FPGA B | 0x1C000XXC | BUS XX Name | The name of the bus XX (schematic) |
| | | | |
| FPGA C | 0x20000000- | DDR2 C space… | Mapped to DDR2 SODIMM… |
| FPGA C | 0x27FFFFFF | … | … interface |

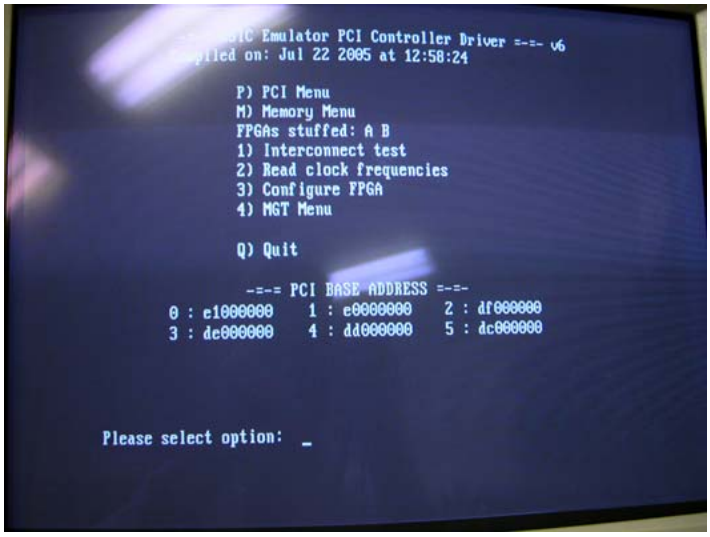
| | | | |
|---|---|---|---|
| FPGA C | 0x28000002 | IDCODE | 0x05000121 |
| FPGA C | 0x28000004 | INTERCONTYPE | 0x34561111 |
| FPGA C | 0x28000006 | RWREG | Scratch Register for testing |
| FPGA C | 0x28000010 | LED_OE | Controls LED output enables |
| FPGA C | 0x28000011 | LED_OUT | Controls LED outputs |
| FPGA C | 0x28100001 | CLK_COUNTER | Contains contents of ACLK counter |
| FPGA C | 0x28100002 | CLK_COUNTER | Contains contents of BCLK counter |
| FPGA C | 0x28100003 | CLK_COUNTER | Contains contents of DCLK counter |
| FPGA C | 0x28100004 | CLK_COUNTER | Contains contents of SYSCLK counte |
| | | | |
| FPGA C | 0x28000001 | DDR2HIADDR | upper address bits for DDR2 interface |
| FPGA C | 0x28000003 | HIADDRSIZE | number of bits in DDR2HIADDR |
| FPGA C | 0x28000005 | DDR2SIZEHIADDR | The size of the DDR2 module. |
| FPGA C | 0x28000007 | DDR2TAPCNT0 | Current IDELAY values of DDR2… |
| FPGA C | 0x28000008 | DDR2TAPCNT1 | …interface |




## 2.1 Using the Reference Design

### 2.1.1 Built-In RocketIO test

From the AETest main menu, select option 4, MGT Menu. The MGT test sends a repeating test pattern out all of the RocketIO transmit pairs, and compares the input of each RocketIO channel to that pattern. To run the test, you must loop back each RocketIO pair.


Emulator PCI Controller Driver =-=- v6
on: Jul 22 2005 at 12:58:24
P) PCI Menu
M) Memory Menu
FPGAs stuffed: A B
1) Interconnect test
2) Read clock frequencies
3) Configure FPGA
4) MGT Menu
Q) Quit
-=-= PCI BASE ADDRESS =-=-
0 : e1000000 1 : e0000000 2 : df000000
3 : de000000 4 : dd000000 5 : dc000000
Please select option: _

You can easily loopback the SMA channels by connecting the RX and TX connectors of each MGT pair together with an SMA cable. The SFP modules can be tested with an LR loopback attenuator.

Option 5 of the MGT menu allows you to invert the polarity of one of the SFP channels. For the test to pass, this must be done, since SFP2 is received with inverted polarity.

The MGT menu also allows you to modify the

The MGT tiles are connected as follows

| | MGT A | MGT B |
|---|---|---|
| COL0, TILE0 | QSE 1 | QSE 1 |
| COL0, TILE1 | QSE 1 | QSE 1 |

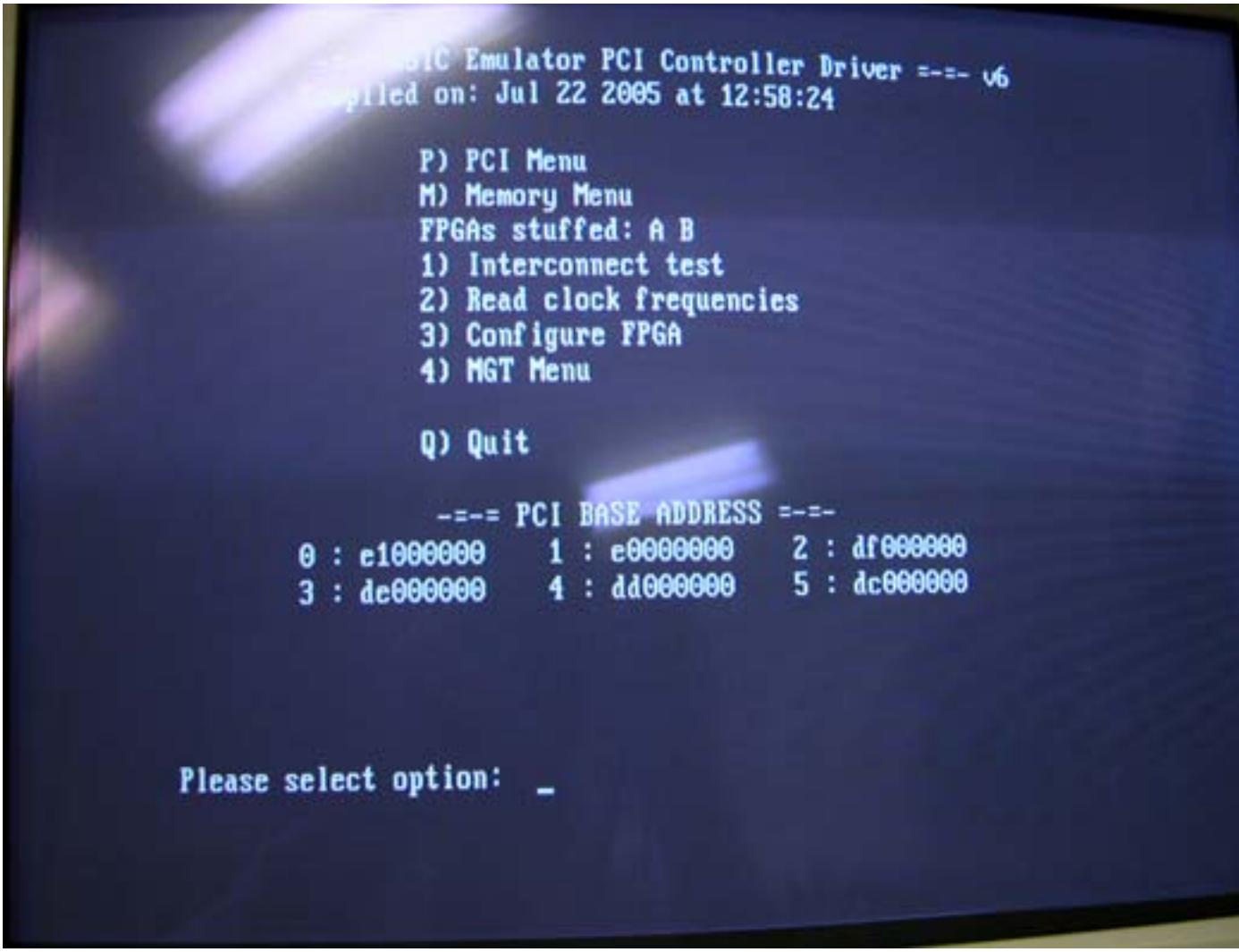
| | | |
|---|---|---|
| COL0, TILE2 | SFP 1 (XFP REFCLK1) | XFP 1 |
| COL0, TILE3 | LOOPBACK | SMA J22 |
| COL1, TILE0 | QSE 0 | QSE 0 |
| COL1, TILE1 | SMA J31 | SMA J25 |
| COL1, TILE2 | NC | SMA J17 |
| COL1, TILE3 | XFP 2 | SFP 2 |

REFCLK2 – 250MHz EPSON

REFCLK1 – ICS 84020 Synthesizer

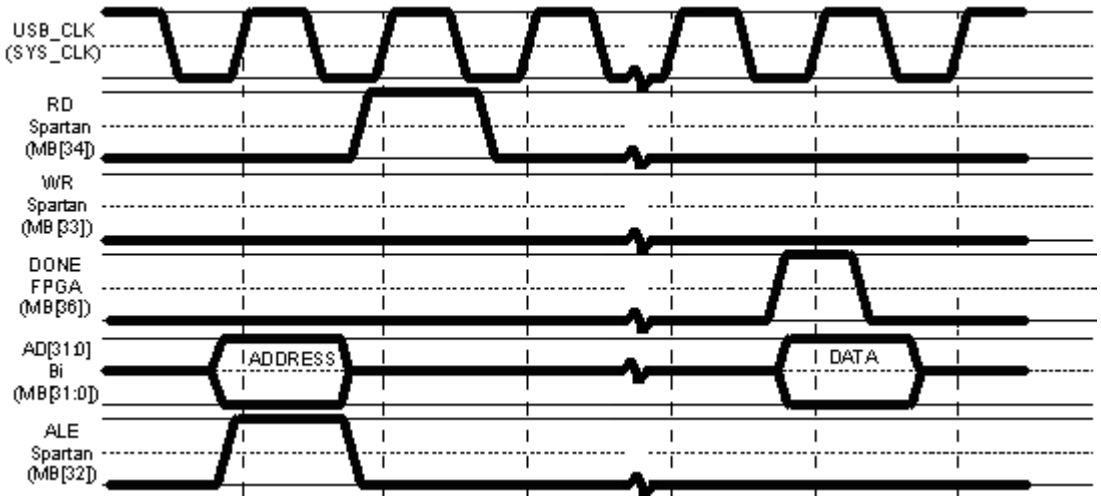
FPGA_A
ACLK/4 18.752 BCLK 132.452 DCLK 92.698 SYSCLK 48.000 SCLK0 0.000
SCLK1 0.000 SCLK2 0.000 RCLK0 0.000 RCLK1 0.000 RCLK2 0.000
RCLK3 0.000 RCLK4 0.000 RCLK5 0.000 RCLK6 0.000 RCLK7 0.000
RCLK8 0.000 RCLK9 0.000 MCLK0 0.000 MCLK1 0.000 MCLK2 0.000
MCLK3 24.000 MCLK4 0.000 MCLK5 0.000 MCLK6 0.000 MCLK7 0.000
Mhz
FPGA_B
ACLK/4 18.752 BCLK 132.452 DCLK 92.698 SYSCLK 48.000 SCLK0 0.000
SCLK1 0.000 SCLK2 0.000 RCLK0 0.000 RCLK1 0.000 RCLK2 0.000
RCLK3 0.000 RCLK4 0.000 RCLK5 0.000 RCLK6 0.000 RCLK7 0.000
RCLK8 0.000 RCLK9 0.000 MCLK0 0.000 MCLK1 0.000 MCLK2 132.452
MCLK3 0.000 MCLK4 0.000 MCLK5 0.000 MCLK6 0.000 MCLK7 0.000
Mhz
FPGA_C
ACLK/4 0.000 BCLK 0.000 DCLK 0.000 SYSCLK 0.000 SCLK0 0.000
SCLK1 0.000 SCLK2 0.000 RCLK0 0.000 RCLK1 0.000 RCLK2 0.000
RCLK3 0.000 RCLK4 0.000 RCLK5 0.000 RCLK6 0.000 RCLK7 0.000
RCLK8 0.000 RCLK9 0.000 MCLK0 0.000 MCLK1 0.000 MCLK2 0.000
MCLK3 0.000 MCLK4 0.000 MCLK5 0.000 MCLK6 0.000 MCLK7 0.000
Mhz
Press any key to quit (clock frequencies refresh every second)
Current reference clock frequency is 48.000000 Mhz

WARNING:
IS TEST WITH ANY DAUGHTER BOARDS PLUGGED IN!!!
Verbose output (0==none, 9==all)?2
Pause on Errors (default: N)? [y/N] y
Loop (default: Y)? [Y/n] n
Random test data (default: N, for walking bit test)? [y/N] y
Drive rest of board to test short circuit? [y/n] n
Enter all FPGA's to be tested, followed by a return
(example: ABDEF): AB
Loop 1 of 1:
Test: drive from FPGA_A
Tested 468 pins (104 single ended) on FPGA_A
Test: drive from FPGA_B
Tested 514 pins (150 single ended) on FPGA_B
finished with testing
ALL TESTS PASSED
press any key:


Emulator PCI Controller Driver ==- v6
on: Jul 22 2005 at 12:58:24
P) PCI Menu
M) Memory Menu
FPGAs stuffed: A B
1) Interconnect test
2) Read clock frequencies
3) Configure FPGA
4) MGT Menu
Q) Quit
-=-= PCI BASE ADDRESS =-=-
0 : e1000000 1 : e0000000 2 : df000000
3 : dc000000 4 : dd000000 5 : dc000000
Please select option: _

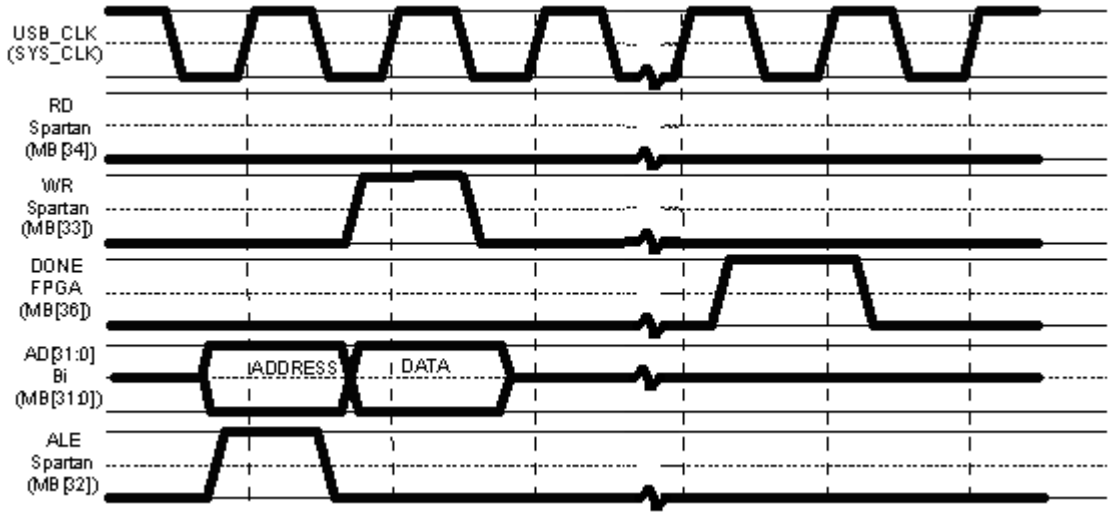
# 3 Main Bus interface

All memory mapped transactions in the reference design occur over the MB bus. This 40-signal bus connects to all Virtex 4 FPGAs and to the Spartan II configuration FPGA. The Configuration circuit (Spartan 2) is the master of the bus. All access to the MB bus (reads and writes) is initiated by the Spartan II FPGA when the reference design is in use.




All transfers a synchronous to the USB_CLK (or SYS_CLK) signal. This clock is fixed at 48Mhz, and cannot be changed by the user. This clock is LVCMOS, single-ended. When the ALE signal is asserted by the configuration circuit, the slave device on the bus (the FPGA) is required to register the data on the on AD bus. This is the "main bus address". All future transfers over the main bus are said to be at this address, until a new address is latched. On a later clock cycle, the master may assert the "RD" signal. Some time after this, (within 256 clock cycles), the FPGA should assert DONE for one clock cycle. On this cycle, the master (Spartan) will register the data on the AD bus, and that will be the read data. If DONE is not asserted, then a timeout will be recorded and the transaction cancelled.

Here is a write transaction:




When the "WR" signal is asserted by the Spartan, the FPGA should register the data on the AD bus. (Note that by convention, FPGAs on the main bus are assigned the address range

corresponding to one value of the highest nibble of the address. Hex addresses 0XXXXXXXX are FPGA A, 1XXXXXXX are FPGA B and 2XXXXXXX are fpga C.)

Some time after this, the FPGA should assert the DONE signal. This will allow the Spartan to begin more transactions. The FPGA may delay this for up to 256 clock cycles before a timeout is recorded and the transaction is cancelled.

Main bus can be controlled from the USB Controller program. (Read and write single addresses, or to/from files) It can also be written from the main.txt configuration method. The main.txt syntax is

MAIN BUS 0x<addr> 0x<data>

Where <addr> and <data> are 8-digit (32-bit) hexadecimal numbers.

# 4 Compiling the Reference Design

The MainTest reference design (for which bit files are included on the user CD and the provided SmartMedia card) can be found on the user CD here.

D:\FPGA_Reference_Designs\
\common\DDR2\controller_ver*
\ddr2_to_mb*
\dn8000k10pci
\LVDS_intercon\source*

The top module is

D:\FPGA_Reference_Designs\dn8000k10pci\LVDS_intercon\source\fpga.v

This module includes all of the other required sources and expects the directory structure found on the CD. (The "LVDS_intercon directory is named that because the "LVDS" bitfiles also found on the user CD and the "MainTest" bitfiles use the same source code with #define statements to switch on and off features).

### 4.1.1 The Xilinx Embedded Development Kit (EDK)

The reference design does not use the embedded PPC processors of the FX FPGA parts. Therefore, EDK is not a required component of the reference design compilation.

### 4.1.2 Xinlinx XST

The Dini Group uses XST software to for design synthesis. The XST projects for each of the 3 FPGAs on the DN8000K10PCI can be found at 'buildxst/*.xst'. These projects have been compiled using XST version 9.1.

#### 4.1.3 Xilinx ISE

Use HDL files as input. Modification of the ISE project may also require modification of the HDL, timing constraints are in files 'buildxst/*.xcf.

A sample Project Navigator project is located at 'DN8000K10PCI/implement/fpga.ise'.

#### 4.1.4 The Build Utility: Make.bat

The Build Utility is found at 'DN8000K10PCI/build/make.bat'. This batch file is used to set system parameters to the desired configuration (i.e. V4FX60 vs. V4FX100, etc.), and to invoke all of the above tools from the command line. Instructions for invoking the batch file can be found by viewing the batch file with a text editor. Additional information about using the batch file to build the reference design is found below. Taking the reference design through all of the various tools for several FPGA's can be very tedious and time consuming- this batch file can do it all in one command!

The command line utility "Make.bat" is an MS-DOS batch file compatible with Windows 2000 and later operating systems. Make.bat should be run from the command line, with command line parameters. It should not be double clicked from the windows environment. A command prompt shortcut is provided in the same directory as Make.bat, and can be double clicked to open a command prompt window with the proper working directory.

Four main steps are involved in building the reference design. Once this has been done once, the Make.bat script can be used to build the netlist with the command Make ppc_netlist. The second step is to synthesize the design with XST. The third step in to place and route, or "implement" the design with the Xilinx ISE tools. The fourth and final step is to compile the PowerPC code and embed it in the bit file. This fourth step is referred to by Xilinx as "updating" the bit file. Hence this fourth step will be referred to as the "update" step.

The build script creates a directory called "out" and places its output files there. After the script completes you will find 3 files for each FPGA that was built. Fpga_*.bit is the file to be downloaded to the FPGA.

# 5 Hardware Test Instructions

The provided reference design is intended as an example code to get you started with the interfaces on the DN8000K10PCI. The provided .bit files on the CD were also used to test the hardware of your board. Some of these tests can be easily repeated by the customer. Bit files are provided for the following tests. All testys except for the PCI test are accomplished using the USBController program on the user CD. This program requires Windows XP.

## 5.1 Main Test

The "Main" reference design refers to a set of bitfiles capable of performing a suite of basic tests on the DN8000K10PCI. The bit files can be found on the user CD in the 'MainTest' directory. To run the test, load these bit files onto the FPGAs and run the USB Controller program. All of

the tests in the MainTest suite require clocks to have certain settings. You can guarantee this by creating a SmartMedia card with the following lines in main.txt

Verbose level:2
sanity check: y
clock frequency: A N 4 M 16 // 100Mhz (not used in maintest)
clock frequency: B N 2 M 28 // 200Mhz
clock frequency: D N 2 M 25 // 200Mhz
clock frequency: 2 N 2 M 24 // 300Mhz

### 5.1.1 Temperature

From the Settings/Info menu, select "Read FPGA temperatures". The temperatures of all installed FPGAs is printed to the log window in degrees celcuis.
PASS CONDITION: The read temperatures should all fall between 20 and 80. "0" indicates a failure to read temperatures, above 80 indicates that the temperature overload circuit failed to reset an overheating FPGA.

### 5.1.2 Main Bus

Make sure that "USB->FPGA" communication is enabled. (The button above the visual display should read "Disable USB->FPGA communication")

From the Mainbus menu, select Read DWORD. Read 1 DWORD from address 0x08000002 for FPGA A, 0x18000002 for FPGA B or 0x28000002 for FPGA C. A message indicating the read value should be returned.
PASS CONDITION: Data should be 0x05000121
Data 0xDEADDEAD indicates a problem with the FPGA configuration
Data 0x12341234 indicates USB Communication is disabled.
Data 0xDEAD5678 indicates an undefined register was read from the reference design.

### 5.1.3 Clocks

From the FPGA Reference Design menu, select "Read FPGA Clock Frequencies". Each FPGA will report the frequency of each global clock measured at the FPGA input.

### 5.1.4 RS232

Connect an RS232 termial to the "user" RS232 port of the DN8000K10PCI. The port settings should be 19200bps, 1 stop bit, 0 parity bits, no flow-control. When the DN8000K10PCI main test is loaded into FPGA A, the terminal should see an echo on the terminal.

This test requires that the RS232 multiplexer is set to FPGA A. This is done from the USB Controller program.

## 5.2 Memory Test

This test requires the "MainTest" bitfiles.

To run the memory test, make sure a 256, 512 or 1024MB DDR2 SODIMM is plugged into FPGA B and FPGA C SODIMM sockets. Select from the FPGA Reference design menu "Test

DDR". In the popup window, select either FPGA B or FPGA C. The status of the test will print in the log window.

This test works by writing a random set of data over the main bus to FPGAs B and C at addresses

0x10000000 – 0x1FFFFFFF (FPGA B) and
0x20000000 – 0x2FFFFFFF (FPGA C).

This memory space is mapped to the DDR2 SODIMM interface. It then reads back the data and verifies it is the same.

## 5.3 Interconnect Test (DC)

This test requires the "MainTest" bitfiles.

The "single-ended" interconnect test tests the DC connectivity. To run the interconnect test, you must select from the "FPGA reference design" menu "DN7/8000K10PCI interconnect". A dialog box will appear asking which type of test. Select "Single-Ended". The status of the test will print in the log window.

The USB Controller has access to the output value, output enable, and input value of each intercoonect pin. The register banks connected to the IO are arranged into "busses". Each bus has an ID code, a OE register bank, a ENABLE register bank, and a IN register bank.

The address of the IO registers are as follows:

FpgaNum (4-bit) | MB_SEL_INTERCON (4 bit) | busnum (20-bit) | reg_offset (4-bit)

FPGA NUM is 0x0 for FPGA A, 0x1 for FPGA B, 0x2 for FPGA C…
MB_SEL_INTERCON is 0xC
busnum is any number, but only low-values (less than LAST_ADDR) will contrain valid busses
reg_offset is 0x0 for REG_OUT, 0x4 for REG_OE, 0x8 for REG_IN, and 0xC for REG_ENABLED

To determine which bits (if any) in a bus are valid, read the REG_ENABLED register. The 32-bits returned '1 are a mask for which of the bits in the REG_OUT, REG_OE, and REG_IN registers are meaningful.

To get the bus ID of a bus, write value 0x1 (32-bit) to REG_ENABLED, then read REG_ENABLED, then write 0x0 (32-bit) to REG_ENABLED. The value returned will be a coded name for the bus. Bits 0-15 are ascii characters representing FPGA names. Bits 16-31 are an arbitrary unique integer distinguishing the bus. Connecting busses from two different FPGAs have the same bus ID.

To cause an FPGA to output signals on a bus, write 0xFFFFFFFF on REG_OE. To set the outputs all to "high" write 0xFFFFFFFF to REG_OUT.

To read the current received value from the bus' inputs, read from REG_IN

## 5.4 Interconnect Test (Highspeed)

This test requires the "LVDS" bitfiles.

## 5.5 PCI Test

This test requires the "PCI interface" bitfiles. This test requires the application "AETEST_wdm"

Run aetest_wdm. The splash screen should print "DN8000K10PCI Virtex- Board VendorID:17DF, DeviceID:1864"

Use the menu items to change A_CLK to 100Mhz. Use AETest menu options to configure FPGA A with the "PCI" design bitfile on the CD. (Blockram_access_A)

From the AETest menu, select "Production Tests Menu". Run the PCI test (0). This test will print a test status.

## 5.6 RocketIO Test

This test requires the "RocketIO V4" bitfiles. (If you are using FX60 CES2 or FX60 CES3 parts, you will require a different set of bitfiles and code. CES1 parts have no working test)

Configure FPGA C with the reference design bitfile. From the settings/Info menu, select "DN8000K10PCI test (B)"

The status of the test will print in the log window. The test will run twice, once for each possible reference clock supplied to the RocketIO tiles. Each tile will print out it's status. Col 0 and Col1 do not share a clock source, so a problem with an entire column may indicate a problem with the clock. (Frequency settings?)

MT stands for the QSE sametec cables.
XFP is the XFP module connector
SFP is the smaller, SFP module connector.
SMA is the SMA cable connectors.
LOOP is a hardware loopback.
COL1 TILE 2A is not connected and should fail.

To check each link, the reference design checks that a certain test pattern is successfully received be the receiver of each tile. If you connect each tile to itself using a loopback on each connector, the test will verify that both the transmit and receive of each tile is working properly.

SMAs are labeled in silkscreen with an arbitrary channel number, polarity and direction. For each channel, connect a cable from RXp to TXp and RXn to TXn. Use two cables of the same length and diameter.

For XFP or SFP modules, connect a LR loopback cable into the module, and insert the module into the socket.

For the Samtec QSE cables, loopback cannot be done unless you have a custom cable. You can test these interfaces by connecting one QSE connector to the next with a  cable. Note that all tiles will not pass this test because one of the QSE has more RocketIO channels than the other.

LOOP should always pass.

Some of the tests (ie XFP) print (ignored) if there is a failure. These tests are expected to fail because XFP connectors are not installed on your board.

# 6 Getting More Information

## 6.1 Printed Documentation

The printed documentation, as mentioned previously, takes the form of a Virtex 4 datasheet and a DN8000K10PCI User Guide.

## 6.2 Electronic Documentation

Multiple documents and datasheets have been included on the CD.

## 6.3 Online Documentation

There is a public access site that can be found on the Dini Group web site at http://www.dinigroup.com/.

## 6.4 Before seeking support

The following mistakes are the most common reason a design does not work on the DN8000K10PCI. Please do the following self-checks before seeking support from the Dini Group.

Make sure that the clock your design uses is running by routing it to an external test point or LED.

Check the pinout in your constraint file. Use the UCF file provided on the user CD for LOC constraints. A netlist of the DN8000K10PCI is provided on the User CD.

Check the .PAR report file to make sure that 100% of your IOBs used have LOC constraints. There is never a situation where pins should not be 100% constrained.

Use the .PAD report to make sure your constraints were applied correctly. Some situations can cause the Xilinx tools to disregard constraints.

Double-check that the connections match between your FPGA pins and the daughter card pins.

If your design uses the MB[37-0] bus, make sure that none of the other FPGAs are driving those MB pins.

Make sure that the "Unused IOBs" option in bitgen settings is set to "Float." If it is set to "Pull down," then those FPGAs are driving any pin that is not assigned in the source code.

If the signals on the SD daughter card are not working, check that you are asserting output enable (OE) for the level-translation buffer.

Chapter

9

# Ordering Information

Part Number

**DN8000K10PCI**

## 1 FPGA Options

### 1.1 FPGA A:

Select an FPGA part to be supplied in the A position. This FPGA is connected to the PCI bus, an expansion header, and can source global clocks. The –12 speed grade is required for full speed operation (1Gbs/pair) of the interconnect between FPGAs.

NONE

LX100 –10 –11 –12

LX160 –10 –11 –12

LX200 –10 –11

### 1.2 FPGA B:

Select an FPGA part to be supplied in the B position. This FPGA is connected to an expansion header, a memory module socket, and can source global clocks. The –12 speed grade is required for full speed operation (1Gbs/pair) of the interconnect between FPGAs.

NONE

LX100 –10 –11 –12

LX160 –10 –11 -12

LX200 –10 –11

### 1.3 FPGA C:

Select an FPGA part to be supplied in the C position. This fpga is connected to a memory module socket. This FPGA is required to provide Multi-Gigabit serial communication. In order to achieve 10 Gbps SelectIO operation, the –12 speed grade is required.

NONE

FX40 –10 –11 -11x –12 (This option makes the 200-pin SODIMM memory socket, one SMA channel, four QSE cable channels, and one optical module socket unusable)

FX60 –10 – 11 -11x –12 (This option makes four channels of QSE cable unusable)

FX100 –10 –11 -11x –12

# 2 Multi-Gigabit Serial Options

## 2.1 Serial Clock Crystals

If you need to interface to a specific Multi-gigabit serial IO protocol, you may need to install a custom crystal as a reference to the RocketIO clock synthesizer. Consult the Dini Group when ordering for availability.

The following frequencies (in Mhz) are standard and Dini Group keeps a stock:

| |
|---|
| 9.8304 |
| 12.890 |
| 14.318 |
| 16.000 |
| 21.477 |
| 24.576 |
| 25.000 |

The default option is 25.000 Mhz.

## 2.2 Module Sockets

XFP and SFP Modules provide 1.0 – 10.5 Gb optical serial communications to FPGA C. DN8000K10PCI has two optical ports, each can be installed with either an SFP or XFP connector. XFP modules operate only in the 9.5-10.5 Gb/s range. Available SFP modules operate between 1-4.25 Gb/s. For 10Gb operation, a –12 speed grade FX part may be required. These parts may not yet be available before.

If you have the FPGA C option, you may select one of the following options.

OPTICAL – SFP, SFP (default)

OPTICAL – XFP, XFP

# 3 Other Options

## 3.1 3.3 V Headers

The DN8000K10PCI can be configured to accept 3.3V input and output on a *subset of expansion header pins*. These IOs are not voltage selectable by the software. You must specify on your order that you would like this option. Select any of the following options. The default option is all 2.5V header IO.

3.3V Header A

3.3V Header B

## 3.2 12V Power

Daughter card supply voltages +12V and –12V are, by default, disabled by jumpers R411 (Header A +12V), R412 (Header B +12V), R414 (Header A –12V), R413 (Header B –12V). This default setting reduces the chance of damage to the Virtex 4 FPGA IO buffers due to user error or careless use of probes. Specify this option to have the jumpers factory installed.

# 4 Optional Equipment

The Dinigroup supplies standard daughter cards and memory modules that you can use with the DN8000K10PCI.

- SE card – 80 signals on .1" pitch headers.
- Mictor Card – 5 Mictor38 headers for use with logic analyzers.
- SRAM module for use in the 200-pin SODIMM sockets of the DN8000K10PCI. QDRII, 300Mhz 64x2Mb
- SRAM module for use in the 200-pin SODIMM socket. 64x2Mb Standard SDR SRAM. Pipelined or Flow through, NoBL available
- RLDRAM module for use in the 200-pin SODIMM socket. 64x16Mb, 300Mhz DDRII
- Flash module for use in the 200-pin SODIMM header.
- Mictor module for use in the 200-pin SODIMM header. (2 Mictor 38 connectors for use with logic analyzer)

The Dini Group can optionally provide the following accessories

- DN3k10SD Daughter card (Provides tenth inch pitch test points)
- DNMictor Daughter card (Provides 5 Mictor connectors compatible with logic analyzers)
- Memory modules for use in the DN8000K10PCI DDR2 SODIMM sockets A and B. (Available Q4 '05)

  - QDRII SRAM 64x1Mb, 300Mhz
  - Flash memory 32x4Mb, 2x4Mb serial flash
  - Reduced Latency DRAM (RLDRAM) 64x8Mb, 300Mhz
  - Standard SRAM, 64x2M (Select ZBT, Pipelined, Follow-through)
  - Test connection module (with two Mictor38)

You may also want to obtain from a third party vendor

- 200-pin DDR2 SODIMM(s)
- SFP modules (for Gigabit Ethernet, infiniband, …)

  IBM part 13N1796 from insight.com $180
- XFP modules

  Intel part TXN181070850X18 from insight.com $692

  XFP heatsink/clip – Tyco part 1542992-2

  -5.2V bench supply for powering ECL-based XFP modules (if required)
- Xilinx Parallel IV cable
- LVPECL oscillators for RocketIO MGT clocking. (The DN8000K10PCI is supplied with a 250Mhz oscillator)

  Epson Part EG-2102CA PECL

Xilinx ChipScope for embedded logic analyzer functionality.